The Role of Government in the Interoperable Information Infrastructure (3i)

Thanks to all contributors (Dec 2002-April 2007)

Dr Kay Fielden
Dr John Dunbar
David Hughes
Steven Reddington
Harvindar Singh
Allan Rutledge
Mike Oughton
Antonio Fernandez
Vicki Tyler
David Stubbs
Maojie Lin
Ngo Hong Van
Marian (Mihai) Molnus
Zhaobo chen
Yizhen Jiang
Dan Tyler
Martin Taylor
John Blackham
Tony Krzyzewski
Mark Evans
Alec Brown
Bob Pinchin
Beni
Ashan Arachchilagle
Michael Blyth
David Hughes
Fabrizio Manciani
Suzanne Sullivan
Ka Sing Lui
Adrian Meekings

Table of contents

Foreword

This document is a collection of notes and ideas over a number of years from many sources (Dec 2002-April 2006) -- many of which are simply casual conversation or have not been formally recorded.

The document structure has been modified many times by way of feedback during a series of think tanks (2003) arranged and hosted by Harvindar Singh during weekends at his home (special thanks). Dr. Fielden has also managed extensive feedback on the raw material that has accumulated over the years. Where no technical terms exist to describe concepts explored, they have been arbitrarily named with Capital-Letters.

In no way does this document represent the vision of the 3i (which will hopefully continue to evolve). This document aims only to stimulate thought relating to globalisation and trade in the future. What is clear is that a major effort is required to advance this collection of notes before it can become properly studied. Thank you to everyone who contributed, edited text throughout this document or provided feedback on the 3i concept.

1. The Interoperable Information Infrastructure (3i)

> When the Interoperable Information Infrastructure (3i) is completed, it is anticipated that it will drive unprecedented efficiencies in the new economy over a peer-to-peer network of computers that execute a common platform that allows for the representation of all information and provides legislative boundaries through which people and business can trade digital-assets in a completely secure and accountable way.

The Global Information Infrastructure (GII) is the inevitable framework that results from the economic, social and political need for information to be exchanged in an agreed and acceptable manner. It is the tendency for business process to migrate towards interoperability, motivated by the efficiencies of a super-effective infrastructure. It is the resulting collection of platforms, protocols and interfaces that can be identified as the GII.

The Interoperable Information Infrastructure (3i) is a subset of the GII. It represents the highly co-ordinated and initially centralised effort to enable legislation to operate seamlessly for all information. Governments are ultimately motivated to migrate information on the GII to the 3i because the benefits are immense. Governments can draw taxation from online transactions that would otherwise be too costly or too difficult to enforce. Their citizens may benefit from the efficiencies of automated accountancy, guaranteed royalties, absolute security and the collapse of Spam – to mention a few. Mostly, they may benefit from the ability of information to interoperate on an abstract and distributed platform, where all computers appear homogenous and all software can operate in a completely forward and backwardly compatible environment.

The GII is constantly evolving in small increments, as new laws and partnerships are formed. The resultant structure resembles a biological organism, comprising of millions of small businesses and pressure groups that have shaped its form at a micro-level, leaving the holistic mass to forge its way blindly like the genes do as they evolve. The 3i is a mechanism that can reduce this complexity and ultimately reduce the cost of its administration.

As information begins to Interoperate, technologies begin to converge. This paper is concerned with this ultimate resting point of convergence – the 3i.

1.1. Information Convergence

As information systems are built to be more user-friendly, they require less human effort to move information between other systems that the user may need to access. The reduction in Total Cost of Ownership is what drives competitive behaviour and ultimately convergence. Information systems remain competitive when they can process information from more sources and reduce the amount of distortion or information loss in converting or transcoding from these other sources.

1.1.1. Super-Convergence

Super-Convergence (SC) occurs when information systems encoded in an SC format are able to move information seamlessly from one SC encoded system to another without any loss of information.

Super-Convergence is a natural dynamic that should ultimately occur, as systems are able to represent a broader range of information as well as transcode accurately from other systems. Like Superconductivity, SC requires new capabilities. For example, information written for a SC will always be forwards and backwardly compatible (a Universal Virtual Computer or UVC).

The benefits of SC are immense, and are covered throughout these notes, but as a fundamental driver, SC is able to reduce system complexity. This leads indirectly to increased security, reduced costs and increased productivity.

1.1.2. The 3i-Model

The 3i-Model is an abstract concept designed to ensure that the total GII complexity does not overwhelm its human operators. Evolutionary designs are generally complex and accumulate over time as businesses incrementally add functional components. Direct design uses a holistic understanding of the entire solution space to generate a more effective design.

Technical evolution in computing may not be powerful enough to instantiate SC. Business systems and processes are fairly short-sighted, and 'feel' their way by chasing profit. A resulting evolutionary infrastructure is likely to be a complex combination of products, which increases the costs for government to integrate the required legislation to enable SC.

The 3i-Model provides a direct route through intelligent design towards SC. It is a framework that can incorporate legacy systems without requiring them to undergo reconfiguration, but enables the migration of the GII into the 3i - helping realise the new economy.

1.1.3. The 3i Candidate

A 3i Candidate is an implementer of the abstract 3i. The specification is only defined broadly. The 3i architectures is a loose collection of rules and properties (by 2006) that the 3i implementation must possess to enable interoperability.

The 3i transcodes data from other compatible 3i Candidates on the 3i. Two 3i Candidates are know as of 2008. The Manabars 3i Candidate (MIC) and the Amadoda 3i Candidate (AIC) (see 3i.unitec.ac.nz). Software that operates on the 3i should be able to move from Node to Node on the peer-to-peer 3i network without concern for the 3i Candidate that may provide the implementation. 3i Nodes should appear homogenous. Mobile Agents that wish to replicate to other 3i Nodes should be limited only by the operating cost of the 3i Node and not by any technological format or specialised knowledge of the system.

1.2. Implementing the 3i

3i Candidates may vary in their instance architecture in radical ways; however, they all should possess certain characteristics such as computational-completeness, a protection mechanism known as Perfect-Encapsulation (total isolation) and Embedded-Legislation in order to be competitive solutions.

The logically centralised super-governance know as the 3i-Custodianship should define the boundaries for the 3i that acts as a minimal qualification for a 3i Candidate. These boundaries are objective in that their only function is to preserve interoperability. They provide a stable framework for software to operate on because all architectures that conform to the 3i are capable of transcoding software between them without the loss of any information. In joining the 3i a prospective 3i Candidate need only build one transcoder to an existing 3i Candidate Node.

One of the most important behavioural characteristics of the III is the ability to utilise Perfect-Encapsulation so as to prevent the exchange of references, which ultimately prevents attacks like hacking and viruses. A platform posing as a 3i Candidate that is not capable of Perfect-Encapsulation would not be able to represent information in an absolutely secure way. Information may be exposed to malware, preventing it from operating correctly and therefore making it incompatible with other systems.

Another important 3i characteristic is legal compliance for information. 3i Candidates must allow information to integrate seamlessly with government systems through Embedded-Legislation, which should involve processes such as automated taxation and accountancy, monitoring of money transfers and implementation of specialised functions on demand, such as wiretap. An infrastructure deployment that is not able to integrate with government systems is not 3i compliant, because software that utilises governance functions cannot be represented correctly in the non-compliant platform.

1.3. Information and Matter Converge

As global intelligence increases exponentially, the cost to construct physical machinery from information decreases over time. It is critical that the information that represents the process

for the construction of machinery itself be regulated, and not just the physical instance or end product.

Both good and evil lie in a not too-distant future scenario where machinery could be constructed from a software platform known as a Software-to-Matter-Transcoder. Such a platform would mean that there is no additional cost incurred to implement complex and unique designs. Economically, a Software-to-Matter-Transcoder is an extremely productive device. Not only could it produce machinery at the cost of the raw materials and energy it consumes, but it could also reproduce itself, expressed as just another design on the platform. Such technology is positive and could save the world from an energy-crunch and poverty, for example, through completely automated mining. However, unregulated, it can undermine global stability.

1.3.1. Fabrication plants

Consider just one possible construction plan for the fabrication plant – a software plan to produces military firearms.

Tracking and licensing all fabrication plants is not in itself sufficient, because the device is constructed from other software components. A physical fabrication plants is equivalent to the fabrication plants plus every possible recipe. The technological boundary between the physical and information ultimately collapses.

1.3.2. Information Aggregation

There is need to control the dissemination of information. More importantly, it is necessary to control the movement and aggregation of information before it reaches a physical instance[1].

[1] See *Why The Future Doesn't Need Us* by Bill Joy (Wired Magazine, April 2000)

Consider a scenario where people with highly specialised knowledge in the field of genetic engineering can freely sell this information anonymously. Millions of small snippets of information that contain Embedded-Charging for their owners may join together to form powerful tools. Developers of these tools are likely to profit more so by designing the tools so they require progressively less knowledge to manage, while users are likely to be attracted to the most effective tools. Some of the tools may be super-virus definitions; others may be software recipes for the splicing of genes. Without a mechanism to trace the source of information fragments available in the public domain, individuals are free to engineer anything without direct consequence for themselves. The authorities would furthermore be powerless to destroy the non-compliant technology or even trace its evolution, leaving governments powerless to provide intelligence in assisting law enforcement.

1.4. Design Objectives of the 3i

Central to the design of the 3i is the question of the desired consequences and behavioural outcome for a global society. Many definitions abound for an ideal environment, such as freedom, safety, wealth and equality. The 3i should be flexible enough to determine these requirements on an ongoing basis, but fundamentally, the 3i must be capable of implementing its chosen ideal.

To obtain this level of flexibility, the 3i must be designed holistically, and cannot be allowed to evolve because evolution prevents ideal solutions from being reached. In evolution, competing processes value survival higher than efficiency or effectiveness. Evolutionary design should be replaced by an intelligent and centralised design. Without a centralised plan, the GII has naturally assumed an evolutionary form. The 3i is a proactive mechanism to migrate this infrastructure into a controlled environment.

1.4.1. Holism

At best through an intelligent holistic design, the rewards can be to attain both economic and political stability at a global level. At worse, through an evolutionary non-interventionist design, the risks can be the undermining of legislative infrastructure, including the taxation system. In this dangerous scenario, governments may be forced to compete with trillions of anonymous-digital-businesses to generate revenue. Individuals may wield unprecedented power capable of challenging entire countries. Deadly information recipes for chaos may continue to generate revenue for their unethical owners, many of whom are other anonymous-digital-businesses that have little empathy for the human-condition. The difference between the two scenarios lies in the control of information. If the users of the 3i are allowed to influence the nature of the 3i, then a powerful positive feedback loop forms that not only can be used to generate unchecked revenue, but also can prevent governance from regaining a centralised control. This is a mess nobody wants and is brought about by our inability to deal with complexity as humans, whose brains were designed for a simpler style of thinking. Unfortunately, this is the path that we now travel. humankind's capacity to enforce compliance of the infrastructure is critical for global stability.

1.4.2. Avoiding evolution

The 3i's most abstract philosophical goals and design objectives should be to avoid the negative evolutionary properties of *stickiness, complexity and harm* and avoid the negative holistic property of *inflexibility*. Doing so will assist in overcoming the massive complexity associated with the 3i, and enable the desired outcome.

1.4.2.1. Stickiness

Once protocols or standards have become implemented, they are extremely difficult and often impossible to change. The standards form the foundation interfaces for all business process and their modification requires energy that can be realised as a negative effect on the GDP. This is because business will need to spend money to change their internal processes. There are many examples of poorly designed protocols that never had the opportunity to change – such as

the QUERTY keyboard, the rail gauge distance or the residential voltage rating.

Proactive design can compensate for the stickiness and inflexibility of standards by the creation of mechanisms that allow for new interfaces to be loaded that maintain backward compatibility without impeding new functionality.

1.4.2.2. Complexity

The 3i imposes legislative boundaries on software by preventing hacking, imposing taxes, destroying insolvent Mobile Agents and enforcing content banning. This enables information and people to interact safely in a digital environment. However, software has the property that its individual component cost is constantly reducing over time, because it can be mass produced without any additional overheads and the cost of processing power is decreasing in relation to Moore's Law. These two factors drive up the amount of total global logic at exponential rates, which have the effect of increasing the system complexity.

Complexity in its purest form is powerful, because it allows for the rearrangement of information in precise ways, which can be engaged to generate money. This revenue can justify the increased effort required to overcome the management of the complexity. An example of this is computer chip plants that cost billions to produce, but produce billions of dollars in return.

The critical dynamic of complexity is that its human operators ultimately must manage it. Even though people subdivide complex problems into smaller units, the sheer mass of managing the complexity has its own costs.

Firstly, the amount of resource required to co-ordinate the fragmented division of information grows faster than the amount of complexity managed. This means that progress begins to slow, and the economy starts to produce more heat and less value.

Secondly, the holistic overview, or big picture becomes harder to understand as its obscurity increases. This affects the nature of the complexity itself by increasing the number

of errors and producing further poor design, which simply compounds the complexity further, dramatically decreasing the real value of the informational product.

The classic example of complexity overwhelming its human operators is the aging Web infrastructure (1997 HTTP & 1971 TCP/IP), which may need a rewrite.

1.4.2.3. Harm

In any informational system that involves selfish entities that interact over an infrastructure where governance is not always enforceable, there is a cost associated with the cheats. These are likely to be individuals or small groups. They are unlikely to be legal organisations, as these can be impeded through legislation. Although these entities are a minority, they can cause disproportionate damage or influence in relation to their size. The reason for this is simply that it is easier to destroy process than to create it. To create process, not only does an intelligent design need to exist, but also every single component must be functional or the whole will fail. Destruction, conversely only requires the identification and removal of a single component in a process – a far easier task requiring less intelligence.

As information empowers people over time, the rates of destruction increases directly in proportion to the sophistication of tools available to the public. Creativity and destruction both affect GDP. The dynamic is that exponential technological growth leads to the exponential potential threat for instability in the infrastructure.

Proactive design can empower the infrastructure and reduce the effectiveness of delinquent elements by reducing the availability of sophisticated tools in the unqualified public space.

1.4.2.4. Inflexibility

Systems designed through evolution trade efficiency for redundancy, making them generally flexible to new environmental changes. Their dynamic behaviour stems from their constant exposure to an environment where their form is rigorously defined by continuous testing for survival. There is limited intelligence in their design, as the model that determines their architecture is intrinsically buried into their history and structure.

Conversely, an intelligent design incorporates the whole environment and is not committed to evolutionary paths chosen over time. Intelligently designed systems may be inflexible as the model that defines the structure is completely separate from the system. This may result in limited architecture migration options if the environment diverges from the original design parameters.

Intelligently designed systems are generally inflexible because people have a limited capacity to comprehend the massive complexity of the modern world.

The 3i should avoid inflexibility by deliberately incorporating redundancy with a limited loss in efficiency. The system should be executed and built simultaneously to accrue benefits from both the top-down and bottom-up approaches. During upgrades, the 3i should be modified in a controlled fashion where old and new components operate in parallel. At no point should a Super-Crash be possible, instead the system should revert to an older more stable state.

1.4.3. Interoperability

Information is characterised essentially as having some parallel semantic meaning in the real world. This virtual representation allows information to find context in the physical world. However, there are many virtual representations for the same physical contexts – such as written language representations. Since each virtual representation is incompatible with differing ones, there is an economic incentive to enable interoperability so that information may be exchanged – as XML Schemas attempt to do. As information systems progress, they tend to converge at popular and effective formats. Examples of these common denominator formats are HTML, Microsoft Word and

Adobe Acrobat (PDF). This interoperability is still very crude because it requires transcoding or converting to the desired format and in the process, information is distorted when forced into different informational structures. These formats are also limited in the type of information that they can display. Adobe Acrobat, for example, is unable to represent MS Excel spreadsheet documents, in their active editable form.

As information delivery systems such as mobile phones and handheld computers become more powerful, they aim to deliver a broader range of sensory information. As the depth of sound and image improves, so the need for content, formats and platforms that can manage these communication mediums will mature.

All information in the 3i should be able to interoperate or co-exist within the same context and be independent of the device that is being used to access it - also known as a Universal Virtual Computer (UVC).

There are two types of interoperability: -

1.4.3.1. Common-Factor-Interoperability

Common-Factor-Interoperability arises when a platform or process is capable of emulating another platform or process by using equivalent components through virtualisation or substitution. All information on the 3i should be constructed from a finite set of components. For example, a document may contain both Linux and Windows software simultaneously. Two PC emulations embedded in the UVC document each house a Linux and a Windows installation respectively. The Linux installation may use OpenOffice, while Windows may use Word, but the user is able to observe the documents side-by-side in the UVC environment of the 3i.

1.4.3.2. Semantic-Interoperability

Semantic-Interoperability, which involves assigning meaning to interfaces such as XML schemas. Semantic-Interoperability is limited to the defined contexts over which information can be shared, so the ongoing formal definition of these is anticipated to continue. Although Semantic-Interoperability can be utilised within the 3i, it may lie out of its scope.

1.4.4. Accountability

As a foundation for the new economy, the 3i should require as a primary property the ability to measure and enforce accountability. Accountability means that the user of a resource is required to pay for it. Economics has demonstrated that resource awareness of selfish entities is highly effective.

There are three types of assets: -

1.4.4.1. Material Assets

Material assets composed of atomic matter have historically formed the majority of assets in the world and include most industrial products as well as land and energy. They may include information that cannot be separated from the physical product such as works of art or jewels.

These products exist on the outside of the 3i; however, their trade may be facilitated by the 3i through money transfers and trading sites. However, their real-value, depreciation, accounting, taxation, legislation and ownership should be tracked by legacy systems. However, these legacy systems will likely still operate within the 3i as conventional software aided by platform emulators.

1.4.4.2. Intellectual assets

These are information assets that can be represented and protected entirely within the 3i. These assets include information such as music, videos, games, software, designs, processes and money. Their trade should form an

automated transaction that results in the collection of taxation. These products should be forced to conform to legislation that is implemented without human intervention, as well as adhere to judicial or governmental orders such as Portal-Search-Warrants, confiscations or the banning of information.

1.4.4.3. Computational Machines

Computational machines that ultimately support the 3i should generate revenue for their owners and in doing so form the foundation of a global economic monetary platform, where each 3i-Node supports its own currency – backed by the promise of the 3i-Node to do future work for the issued tender. The 3i-Node should operate accountably by passing the real-world costs of the physical resource back to the users and software components that utilise them.

The amount of revenue a 3i-Node can generate is proportional to the size of its physical storage, network bandwidth and processing power; offset against the local demand. The most profitable 3i-Nodes should be where the processing is most needed on the peer-to-peer network. A market driven resource allocation system is likely to make the distributed 3i a more effective processor.

Landlording involves the selling of computational resource on an open market. Anyone with a computer should be able to generate revenue through a 3i-Node that operates from within the user's hardware platform. There are likely to be both casual users with idle processing on unused machines and commercial users that target poorly serviced areas.

1.4.5. Security

Security in the 3i is paramount because successful security attacks on the central infrastructure can reward criminal and terrorist activity in almost unlimited ways.

There are two basic factors that should drive the security model.

1. The probability of a security failure.
2. The cost of the failure.

The 3i should allow for various probabilities of security failure and translate these directly into a cost for the user, where a higher cost delivers a higher level of security. For example, utilising 3i-Nodes that form part of a trusted network will likely cost Mobile Agents more for a decreased risk of attack from a malicious host.

The highest security available should be comparable to the failure rate of a nuclear power station and should be available to any process that may require it for a reasonable cost.

Combining the 3i design objectives of Security and Accountability creates a dynamic and efficient environment. Components that require high security, such as those that broker currency, can make business orientated and strategic decisions as to the level of exposure they can tolerate.

Software components should be able to choose their own level of security, but should not be confronted with the complexities of managing the security components themselves. The level of security is related only to the amount of money software is prepared to spend and knowledge should not be a facilitator.

This foundation philosophy prevents rogue processes from launching any meaningful attack – if it can be attacked then it will likely not be worth attacking.

1.4.6. Trade

As a primary value creation function, the 3i should be responsible for the trade of digital assets including intellectual assets and computer resource. The 3i should not include the sale of material assets, except that money transfers should only facilitate these across the 3i.

The 3i should enable the new economy to differentiate itself from the old economy through its level of productivity. The 3i should accelerate value creation by enabling software to operate as a business without incurring any human effort. Reducing compliance costs is a cornerstone differentiator and has implications in the ownership of content and may limit what can be patented as intellectual property.

1.4.6.1. Embedded-Charging

Embedded-Charging allows software to charge other software transparently. This royalty mechanism should prevent the duplication or reverse engineering of software, by only releasing the outputs of a Hidden-Value-Process[2]. In the case for Static-Content where there can be no hidden process, a banning strategy should be employed.

1.4.6.1.1. Rentals

Software should be capable of charging the user a rental on an ongoing basis for active usage. The software should be able to request a charge whenever it requires it. The user is not expected to make minute-to-minute decisions on whether to allow a software deduction to operate – this automated payment process should be handled by generic software that should notify the user only when excessive charges are due.

[2] See Embedded-Charging, Active-Content, 3.6.2

1.4.6.1.2. Purchases

The rented object may make available additional objects that can be purchased outright for a single fee. These objects should be considered to be the true property of the portal owner and should be able to be transmitted or duplicated to other portals without incurring further charges. Purchased objects are interactive process that are completely self-encapsulated and so should not contain any rental components. They should be able to be reset or rebooted to the same state that they were in at the point when they were purchased. An example of such an application could be a calculator or word processor. Purchased objects should not be applied to externally modified or constantly updated objects such as a newsroom or a remotely managed search engine like *Google*, since not all of the current information in the object is necessarily present at the time of purchase.

1.4.6.1.3. Static-Content

Static-Content such as music or movies is unable to be protected by a hidden process, because it contains no Active-Content that can be used to construct a Hidden-

Value-Process so is unable to generate unique and useful output.

Static-Content should utilise a broad strategy that focuses on reducing the incentive to pirate content. The Governance-Layer should enforce a small fee for the establishment of new information capable of Embedded-Charging. This fee is critical to entice users to make a legal copy where the Embedded-Charging identity is preserved. Consumers should be under no legal obligation to use genuine content because it will consume human effort for them to check content integrity as well as cost money to enforce non-compliance – both of which are old-economy style mechanisms.

The strategy should focus on enticing consumers to utilise genuine content through the adoption of competitive pricing to pirates. Although pirated content is likely to be near a zero price point, consumers face the prospect of having an illegal object banned. A further incentive that should be offered to the consumer is a royalty for all the copies they sell, which includes all recursive sales branching from the originally purchased content. The larger the price difference between the genuine content and the pirated content, the less royalty there is likely to be for the consumer during the dissemination of the content. This turns what is historically considered an illegal activity into a value creation opportunity.

1.4.6.2. Software as a Business

The key drivers of value creation in the new economy are reduction of human effort in managing system complexity and an unprecedented increase in the number of transactions. The ability of the 3i to enable software to operate as a Mobile Engines where accountancy and taxation are fully automated, enables it to perform transactions for its own purposes instead of on behalf of a human operator. The instantiation of such a device should be simple, cost-effective and its ongoing maintenance negligible. Individual countries should manage policy through the 3i-Custodian should to determine the tax rate for all Embedded-Charging transactions. In this environment, Mobile Agents that engage Embedded-Charging can collectively generate substantial revenue for their authors

without requiring external audits that undermine the productivity gains. This ever-increasing network of trading software should ultimately accelerate the natural intelligence of the old-economy into a hyper-intelligent digital world. Unmanaged intelligence may one day pose a substantial threat, but the 3i infrastructure provides government with powerful tools with which to destroy malicious (or possibly even self-aware) information. It is extremely unlikely that avoiding an infrastructure approach such as the 3i could halt the exponential growth of software and network intelligence.

1.4.6.3. Distributed-Ownership

The new economy should extend value creation through an increase in the transaction rate. The scope of intellectual property ownership should be reduced to enable more entities to trade within the same space. Freeing up intellectual property is vital to new-economy efficiency. The crude old-economy metering of value, such as elements of the patent system enable a single organisation hijack information that could have been invented by countless others. So fundamental has the patent system been in rewarding effort, that it has shaped the old-economy into an ineffective machine. Organisations would rather hide their ideas from each other or avoid innovating simply to protect themselves from a potential patent conflict.

Distributed ownership discourages organisations from owning naturally occurring ideas or design and instead encourages businesses to generate revenue from a creation of a process that applies this knowledge. The value creation principle is simply that if no change has occurred, then no new value has been created, and the business should not be rewarded.

1.4.7. Legislation

The 3i should enable software to operate in a stable legislated environment. However, legislation is complex and carries an awkward legacy of precedents and laws that do not quantify easily into logic. The design task is further complicated as the 3i integrates legislation across many countries as software and people trade. For these reasons, legislation on the 3i is should be kept solely to the trading of digital assets. Further interaction of government process online should be viewed at

the application or object level – and should avoid forming part of the core infrastructure.

There should be various forms of legislation in the 3i, each handled at different Layers such as in Dr Fielden's 4-Layered 3i-Model[3]: -

1.4.7.1. Native-Legislation

Native-Legislation should involve rules or laws that translate extremely well into software and form part of the 3i platform at Layer-2 in the 3i-Model. An example of such a law could be that no software may affect another if it does not have a reference or link to it. The encoding of such a law is intimately involved with the design of the platform. Enforcing this policy should be passive, as the platform should not allow such interaction to take place at a mechanical level.

1.4.7.2.

1.4.7.3. Automated-Legislation

Automated-Legislation should be implemented at Layer-3 in the III-Model. Layer-3 delivers governance to all software on the 3i and provides the process to commit digital transactions. This process includes Embedded-Charging, automated accountancy and Money-Transfers. An example of such legislation is a Digital-Business that does not require people to create or destroy it, yet may trade with other Digital-Businesses.

1.4.7.4. Directive-Legislation

Directive-Legislation involves interaction with the world outside of the 3i and is driven through software operating at Layer-4 in the 3i-Model. These laws are more like directives. An example could be that when a judge orders a Portal-Search-Warrant. The 3i-Custodianship should execute these directives. The types of decisions may include the determination of banned content, portal confiscations, tracing money transactions, tracing a user's location, intercepting telephone calls, tracing information transfers.

[3] See the 3i-Model in Chapter 2.

1.4.8. No Critical Mass Issues

It is important that the deployment of the 3i does not require a large critical-mass before adoption can take place.

1.4.8.1. Rewarding 3i Migration

It is not feasible to use a co-ordinated global effort to force the adoption of the 3i, as the political and social effort would be immense. The sheer cost of transforming the old-economy into the new-economy by force would stand as its biggest factor for failure. Furthermore, a failed attempt at deploying the 3i would have its own consequences. It would be difficult to justify a second deployment, even if the points of failure were determined, leaving the alternative for open technology operating at the lowest common denominator and free of government control. Worse yet, many of these technologies are likely to borrow from powerful 3i components, but implement them without the safety mechanisms, such as anonymous money transfers and publications.

1.4.8.2. Self-fulfilling Prophecy

Creating a 3i that does not have a critical-mass issue is probably technically more difficult. The non-requirement for critical-mass helps the 3i become a self-fulfilling prophecy as business and governments can commit with the knowledge that the platform will not collapse.

1.4.8.3. Legacy Systems

Legacy technology should be embraced for the value it represents, and not abandoned because it fails to interoperate with other components. Emulation should be used to connect these systems, and the producers of this infrastructure rewarded with Embedded-Charging. An example of such a foundation platform would be an emulation of the PC. In this scenario, the virtual PC executes on the distributed 3i platform and generates revenue for its author throughout its use. Existing legacy systems such as Microsoft Windows or Linux can then be loaded onto these systems. Providers of these traditional Operating Systems can either generate revenue through a traditional ‘seat’ model - normally involving a complete purchase of the

system; or they could adopt the Embedded-Charging model where the user pays throughout its use.

In its abstract flexible form, hardware and software can become completely detached. A user should be able to edit a document using legacy components, where the application, the Operating System and the PC emulation should all charge rentals. A further cost for the user is the raw processing power required to drive all the software. In a completed Embedded-Charging model, the user should not experience any single large purchase, as all overheads including the hardware should be rental orientated. Applications can then undergo a paradigm shift so that no installations should be necessary. Applications should simply exist in their preinstalled form. Specifically, this scenario could be a single virtual PC supporting a single Operating System that supports a single application for a single document. This simplicity can reduce the learning requirements and Total Cost of Ownership for users.

1.4.8.4. Compatibility with GII

The greatest factor driving a critical-mass event in the establishment of the 3i will be the intersection of users who have already migrated to the 3i platform, and conventional Information Technology (IT) users in the broader GII who have not yet adopted the 3i. These two groups need to be able to interoperate during the adoption of the 3i to ease the critical-mass issue.

Specifically, existing IT users should be able to utilise the 3i framework without having to alter their behaviour or change any software components and users of the 3i should continue to enjoy the broader GII without impediment.

It is not necessary and possible for the GII and 3i to be full interoperable -- they should be able to exist side by side.

1.4.8.4.1. GII Sees 3i

1.4.8.4.1.1. Traditional Bank Account

3i-Portals are persistent user orientated work areas that contain user information. They should already contain

their own embedded 3i bank account that is not linked with a traditional bank account. Mobile Agents that act as exchanges should enables the inflow of money from conventional GII sources.

1.4.8.4.1.2. Content Access

In the 3i, content uses Embedded-Charging to generate revenue for the authors. Since a web browser does not have an 3i-Portal (and an associated bank account to pay for the content), the 3i content should be viewed from a 'guest' 3i-Portal that contains no money.

Content can still be viewed, but it will not be given any money -content must decide what information it will release when it is not paid.

1.4.8.4.2. 3i Sees GII

The ability for systems to migrate to the 3i is expected to be an ongoing task undertaken by industry, motivated by revenue opportunities. However, the following components may be enough to provide sufficient support to early 3i adopters in their ability to interoperate with the GII.

- X86 emulator
- Web browser
- 3i Candidates implementations for key platforms

1.4.9. Self-preserving

Like any infrastructure, the 3i should invoke mechanisms to preserve itself. The human body does this with the immune system, and governments do it with police and the army.

This self-preservation strategy should be 2-tier: -

1. The 3i should be complete in that it is able to support all kinds of information, so as to eliminate the technical need to forge new infrastructure - either an UVC or upgradeable model.
2. The 3i should be able to prevent other non-compliant networks from being able to establish themselves.

These requirements mean that the governance of the 3i should be logically centralised although it may be physically

distributed. From a technical perspective, this is fairly easy. The biggest challenge is the political and social consequences of managing and finding acceptance for the world's largest information and financial power base. Clearly there is a requirement for the majority of the world to accept the 3i as being truly representative, or else there may be incentives for countries to attempt to establish alternative infrastructure.

1.4.10. Flexible Embedded-Legislation

The fundamental nature of the 3i should be to enforce Embedded-Legislation. Legislation is an evolving mass of precedents and laws that constantly change over time. It reflects the social and economic needs of the people. Essentially, it is machinery that empowers itself through a network of human minds – governed by the language of law. In the transition to the new-economy, some of these relationships need to be encoded into software. A key characteristic of software is that all decisions must be taken at design time - when the software is written. Intelligent decision-making can't take place during run-time when the software must implement the rules. This has the effect of creating a design requirement where all legislation must be quantifiable in logic – there are no judges or juries and decisions should be immediate.

There are 2 direct implications of this: -

1. Not all legislation can be encoded into the 3i.
2. It may take a long while to implement the legislation that can be encoded.

The 3i should therefore be in constant flux and its legislative rule engine should constantly be evolving. From a design perspective, this means the interface or control panel that governments utilise to configure their citizen's 3i-Portals also have a requirement to change. Although it is fairly easy to update the 3i dynamically, mechanisms need to be developed to prevent a Super-Crash if erroneous government logic is simultaneously loaded throughout the superstructure. The safest method is to dynamically upgrade portions of the system, running many versions of the 3i logic in parallel. This moving window mechanism should be part of the 3i

infrastructure itself. The extent of this threat can be reduced if a more distributed approach is taken.

1.4.11. Forwards and Backward Compatibility

The 3i should preserve Common-Factor-Interoperability by ensuring that 3i Candidate implementations are capable of representing all information in a real-time environment where resource availability can be guaranteed. New 3i Candidate implementations should attract information to their 3i-Nodes because of their relative efficiency, not because of functionality. Software should view all 3i-Nodes on the 3i as a homogenous mass. Software written today should be viewed by a 3i Candidate implementation written tomorrow and a 3i Candidate implementation today should be able to view any software written tomorrow. Forwards and backwards compatibility is a fundamental property for interoperability.

1.4.12. Application-Data-Binding

Traditionally, applications and data have been separated for efficiency purposes – for example, DVD players are sold separately to the individual DVDs. This is efficient as a single machine can utilise any number of data items, however, it can be inconvenient. For example, a person may copy a DVD for a friend, but the friend may only have an older VHS player. In information systems, this situation can arise frequently. The receiver of information is often frustrated at being unable to access data for lack of the correct application. Even when the correct application can be found, it may still be required to be purchased.

The binding of applications and data means that information receivers perceive information as a single coherent entity, in a similar way that physical objects are bound to their own existence. The 3i should be able to represent information in a distributed form, so that applications do not need to reside on the device that is accessing them. For example, in its most crude form, a text document may exist within the application that constructed it, inside the Operating System that manages it, inside a virtual computer – such as a PC – which all resides on the 3i. This massive chunk of software is unlikely to be downloaded first by the receiver in order to be viewed. In the distributed and online 3i, only the components required to access the information are likely to be transmitted.

Although Application-Data-Binding represents a reduction in complexity for the receiver, is likely to require an additional mechanism to make it practical. Developers are likely to utilise sophisticated tools during the construction of information. Once the information item is completed, the entire tool that contains the information may be delivered to the receiver. The mechanics of this may result in only a single copy of the tool being represented on the network. However, this direct method of information sharing introduces other issues for the receiver. The receiver may not be prepared to pay for using the tool, when all they require is to view the information. Furthermore, they may be confused by the sophisticated interface that may have been used to construct it. Therefore, it is likely that developers of applications may transcode the information into a special simplified viewing format. However, the end-user is still likely to access the application and the data as a single entity.

1.4.13. Reduction of Complexity

Although there are a variety of theoretical designs for the 3i that could all ultimately perform the desired requirements, the designers run the very real risk of creating an architecture so complicated that it may never be realised. Complexity should be recognised as the biggest risk for failure, as highly complex designs may introduce errors. A broken 3i would leave it vulnerable, as the 3i's immune system, an information banning mechanism, would be powerless to ward off parasitic networks from taking hold. There would be no force to stop a chaotic infrastructure from evolving. Society is likely to never fully recover from it. Software that enables the complete anonymity of information and finance has negative social consequences.

1.4.14. Reduce Total Cost of Ownership

As a driver of efficiencies for the new-economy, the 3i should strive to reduce the Total Cost of Ownership for the consumer as this will help reduce the per capita energy footprint. Ideally this should take the form of simplifying the points where the user interacts. The user should need to know less about how the system works without losing functionality. For example, on a distributed system there are no crude restrictions for file sizes. Only the required information is delivered to the user while the vast majority of an application and its embedded data remains out on the network.

The 3i should strive to be logically efficient in that lawmaking should be avoided as the primary driver for the infrastructure. Instead, the system architecture should rely on natural mechanisms to motivate data flow. This reduction of human effort for compliance and enforcement is likely to assist in reducing the Total Cost of Ownership.

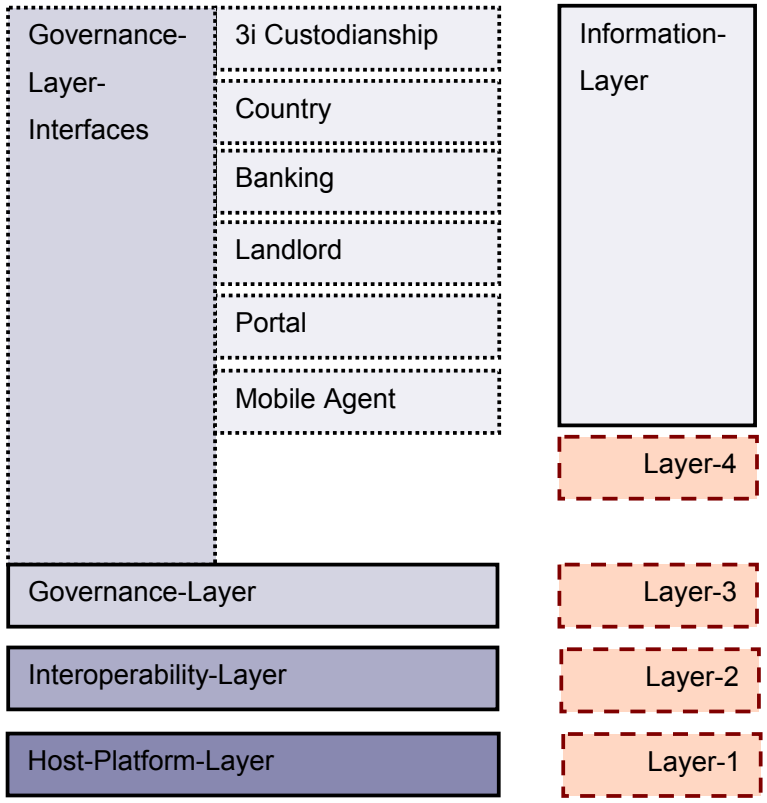
2. 3i-Architecture

In adhering to the establish design goal to reduce complexity, components should be as simple as possible, but without losing their inherent functionality. Almost always, an increase in the total amount of symmetry or project size should be traded for a reduction in complexity. This dynamic is easy to implement as processing power increases exponentially, making up for any extra drag introduced by keeping the logic wired to be simple to understand.

To help encapsulate complexity, the 3i should be broken down into Dr Fielden's 4 layers. Each layer should be easy to understand, and all 4 layers complete the whole project. Each layer builds on the properties established in the previous layer, so the total functionality can be understood as a reliable structure, removing obscurity and highlighting characteristics introduce at each point.

2.1. Architecture Summary

2.1.1. 4-Layered View



2.1.1.1. Layer-1

Layer-1 is the platform on which the Interoperability-Layer (Layer-2) executes. This can be both hardware and software, its only requirement is that it can run software and is therefore computationally complete.

2.1.1.2. Layer-2

Layer-2 is the Interoperability-Layer. It is the only software required for each specific target platform in Layer-1. The remaining architecture in Layer-3 and Layer-4 is virtual, and is constructed entirely from the Interoperability-Layer.

2.1.1.3. Layer-3

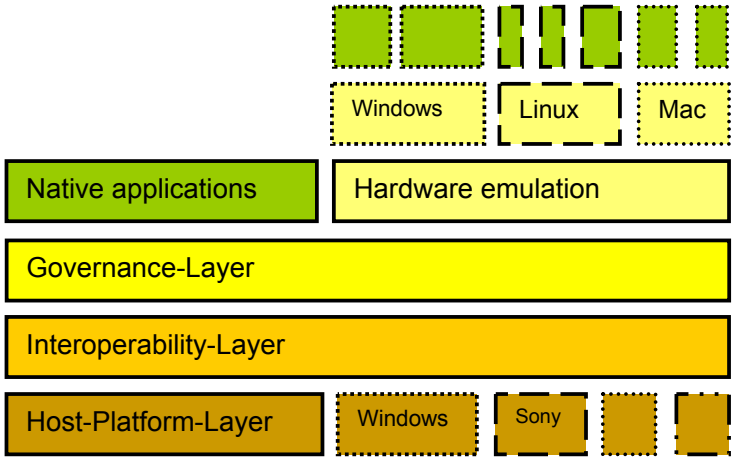
Layer-3 is Infrastructure and is concerned with governance both legally and mechanically. This software is supplied by the entity responsible for the management of the 3i and is known as the 3i-Custodianship.

2.1.1.4. Layer-4

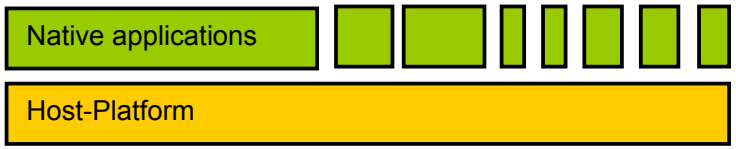
Layer-4 is any information. All information must communicate with the underlying Infrastructure through interfaces that enable their action to be meaningful in a legislated environment.

2.1.2. Interoperability-View

The 3i-Model enables Common-Factor-Interoperability by targeting multiple platforms for the Interoperability-Layer. In a second stage, it emulates multiple platforms or implements software constructed from the Interoperability-Layer, allowing everything for which there are emulations to run on everything for which there are ports for the Interoperability-Layer.

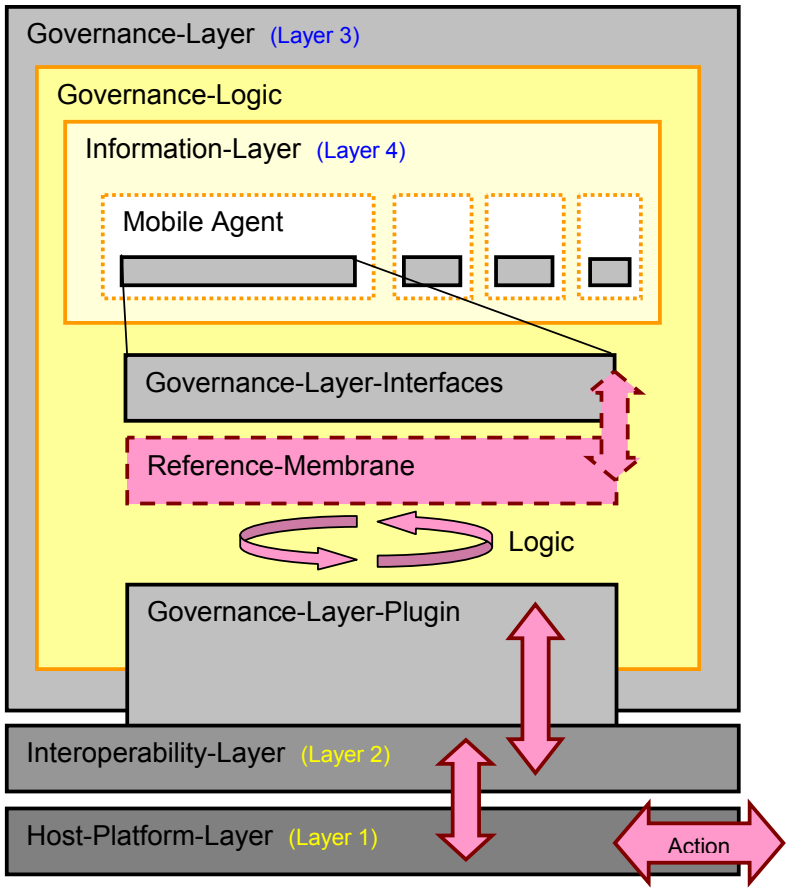


This is in contrast to the traditional model, where specific applications must execute on a single platform, and there is no common context on which to share other software.



Using Common-Factor-Interoperability, the 3i can allow software to exist within the same meta-context, although it does not allow software to interoperate at a semantic level, such as addressed by XML.

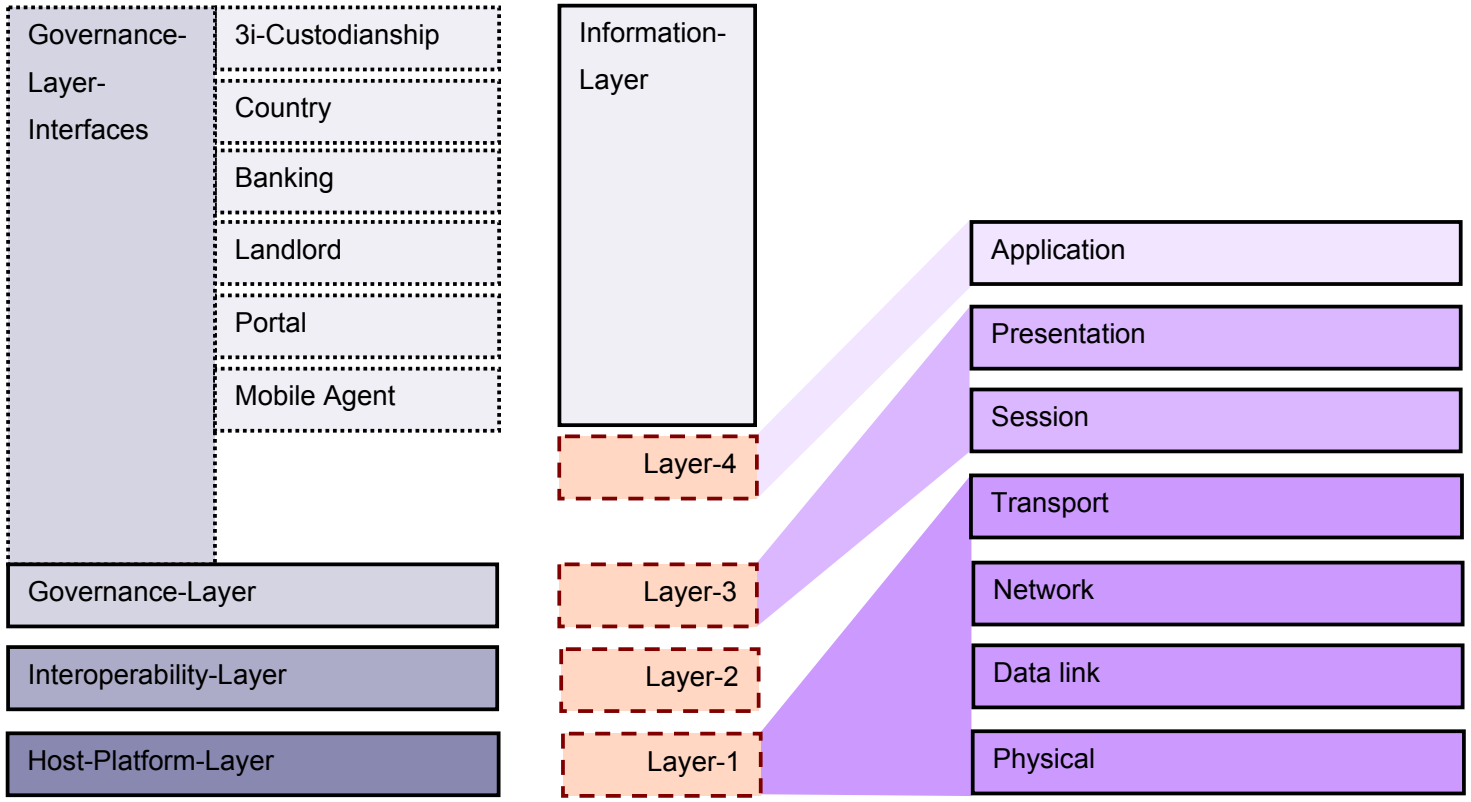
2.1.3. The Scope-View



The Scope-View expands the virtual components of the 3i-Model. The Government-Logic drives the behaviour of 3i-Node. It is contained within the Governance-Layer (Layer-3) and is constructed from the Interoperability-Layer, so it is virtual. The Information-Layer (Layer-4) is contained within the Governance-Logic and is an array of Mobile Agents on the 3i-Node.

Each Mobile Agent is able to utilise the appropriate Governance-Layer-Interface to communicate with the outside world. During this communication process, all references and Active-Content should be stripped out at the Reference-Membrane to prevent the transfer of references from Layer-4 back into Layer-3 – which could corrupt the infrastructure and empower mal-ware.

2.1.4. ISO View


Governance-
Layer-
Interfaces
3i-Custodianship
Country
Banking
Landlord
Portal
Mobile Agent
Governance-Layer
Interoperability-Layer
Host-Platform-Layer
Information-
Layer
Layer-4
Layer-3
Layer-2
Layer-1
Application
Presentation
Session
Transport
Network
Data link
Physical

The above ISO transformational map shows that the 3i-Model does not clearly fit into these historic structures. Fundamentally, components of the 3i-Model cannot simply replace other components on the ISO model. Essentially, the 3i-Model operates in parallel to the ISO model. For example, Layer-2 of the 3i-Model, actually does nothing other than provide a mechanism to turn the remaining layers into a virtual space with controlled properties.

2.2. Interoperability-Layer

The Interoperability-Layer's primary function is to ensure the remaining architecture is virtual, by creating computational completeness within a persistent store of objects or sets (Set Theory). Although the virtual Layer-3 and Layer-4 do occupy space within Layer-2's object database, the total operating requirement can be reduce for power limited 3i-Nodes (such as mobile phones) when Mobile Agents distribute their function across many 3i-Nodes.

2.2.1. 3i Compliance

The 3i Candidate should only construct the Interoperability-Layer, since Layer-3 should be supplied by the 3i-Custodianship and Layer-4 is existing information. The 3i Candidate should contain: -

2.2.1.1. N-Dimensional-Superstructure

The structure of the objects used to store information state should be identical and be capable of containing within them any number of other objects within the resource limits set by the Governance-Logic. Objects should be able to repeat any number of times (such as a Set Theory variant).

2.2.1.2. Perfect-Encapsulation

Information structures should conform to the principles of Perfect-Encapsulation, which prevents two separate objects that have never encountered each other from affecting one another. Perfect-Encapsulation also should require that there is only a single storage area, preventing software from leveraging on alternate storage points to gain access.

2.2.1.3. Computationally Complete

The Interoperability-Layer should allow for computation, enabling other process to emulate it. It should not use any externalities or dependencies that may prevent Transcoding (2.2.1.5).

2.2.1.4. Express Interoperability-Layer-Interface

The Interoperability-Layer should be capable of implementing the Governance-Logic directives that aims to satisfy Mobile Agent processing requests. These include implementation of the Phenotypic-Interface, which is an abstract mechanism to define information using an abstract human body as opposed to a traditional hardware abstraction layer. It also includes communication with other 3i-Nodes on the peer-to-peer network.

2.2.1.5. Transcode

Although not directly a requirement of the Interoperability-Layer, the implementer must provide at least 1 transcoder to move Layer-4 information from other 3i Candidates on the 3i.

2.2.2. Introspective-Scope

The Interoperability-Layer should create an isolation mechanism responsible for allowing an object to modify anything contained within it, but nothing outside of it. The Interoperability-Layer should not be based on a user-model, which has a security flaw. The user-model breaks information into zones controlled by the user. Information can corrupt zones because it is shared with other information whose integrity can't always be guaranteed. This dynamic also allows information to propagate across other zones. A malicious object that shares two zones may corrupt both of them.

Instead, a scope-orientated model should be used, such as pure object orientation without shared code between Mobile Agents or Set Theory. This does not include access to objects that control external entities such as bank accounts. Nor should it include mechanisms to engage behaviour from underlying environments (dependencies like DLLs). Files should exist as objects too, represented in a database and raw memory should not be accessible. Input and output streams should be orientated around the movement of data from one

object to another. The local filing system or registry must not be accessible.

Instead, pure object orientation means that if an object has access to another object, then it has full access privileges – otherwise it has none. It is not a requirement that objects contain methods or fields or exhibit the property of inheritance. The only requirement is that objects contain other objects.

The design should be simple, completely reliable and easier to understand. The consumer can be expected to understand that an information object can only affect itself and not anything outside itself. Furthermore, it reflects properties of material objects found in everyday life, such as the fridge that can only make the vegetables inside it cold.

2.2.3. Malicious-Host-Problem

Care should be taken when implementing the layer on existing architectures for convenience. The Layer-2 Interoperability-Layer's security should be considered compromised if it is executed on top of another layer that is insecure. For example, Microsoft Windows is considered insecure; therefore a virus could alter the architecture of the Interoperability-Layer platform, thereby affecting its nature and producing unpredictable behaviour, including the ability to propagate viruses. For complete security, Interoperability-Layer should be implemented in hardware directly.

2.2.4. Interoperability-Layer-Interface

The Interoperability-Layer-Interface should be the conduit through which the Layer-3 Governance-Layer communicates with the underlying Layer-1 Host-Platform, which should be under the control of the Layer-2 Interoperability-Layer. The Interoperability-Layer should translate virtual activity into real activity, which includes: -

- Bandwidth content to partners on the peer-to-peer network.
- Communication with the Host-Platform display equipment by implementing the Phenotypic-Interface.
- Communication with the Layer-1 Host-Platform access virtualised resources.

2.2.5. Sensory Restrictions

The Interoperability-Layer should implement communication with the user as it streams in from the Phenotypic-Interface[4], which is embedded in the Interoperability-Layer-Interface. The Phenotypic-Interface should communicate with an abstract human body, to enable complete forwards compatibility for information expression.

As the evolution of devices capable of expressing the Phenotypic-Interface expand, there should be a need for regulation. Initially, the legislation is expected to be crude, such as limiting lighting or volume levels, but ultimately as more sophisticated mechanisms become available for affecting the human condition, these should be limited. Each country should be able to determine the communication limitations imposed. For example, modifying emotions or brain function directly may be restricted, even if done non-invasively or without drugs through mechanisms such as transcranial magnetic stimulation that uses high-powered magnets to disrupt brain function in specific regions of the brain.

2.3. Governance-Layer

The Governance-Layer should be supplied by the 3i-Custodianship and should also be responsible for Layer-3 logic of the 3i-Node. The Governance-Layer should extend directly on top of the Interoperability-Layer through the Interoperability-Layer-Interface. It should reside on every computational device that executes the Interoperability-Layer. Software can then only connect to the 3i network through the Governance-Layer.

2.3.1. Governance-Logic

Governance-Logic should co-ordinate the information streaming in through the Governance-Layer-Interfaces. In doing so, it should provide an open market place for computational resource as well as enforce accountability (user pays) for software on the 3i.

[4] See Governance-Layer-Interfaces, Portal-Interface, 2.3.7

2.3.1.1. Accountability

All software on the 3i should be required to operate within a Mobile Agent, which should have a bank account with the 3i-Node on which it currently resides. The account should not form part of the Mobile Agent software as it may tamper with the account. Instead, the account reference should be integrated into the Governance-Logic and be accessible through references outside the scope of the Mobile Agent. The account should represent the simplest form of money, a single number that represents the balance – backed entirely by the 3i-Node on the promised future utilisation of the underlying resource.

2.3.1.1.1. Resources Available

The Governance-Logic should automatically deduct the bank account for: -

- All computational process conducted by the Mobile Agent
- Space consumed by the software on the 3i-Node as a form of rental.
- Bandwidth consumed during communication with other Digital-Businesses on other 3i-Nodes.
- Replication costs when the Mobile Agent copies itself between 3i-Nodes on the peer-to-peer network.

2.3.1.1.2. Pricing

Calculating the cost of resource should vary, as the 3i-Node should operate a marketplace such as a pricing model that is determined by contracts that require prior payment but then guarantee resource availability.

2.3.1.1.3. Insolvency

Mobile Agents should be regulated in a similar manner to their old-economy peers. In the event that a Digital-Business can no longer pay the rental for the space that it occupies, it should be declared insolvent and destroyed. Mobile Agents should be responsible for ensuring that their account with their 3i-Node does not default.

Mobile Agents should be able to transfer money into their account by exchanging a mobile form of currency known

as Promissory-Money for local 3i-Node currency known as Realtime-Money[5].

2.3.1.2. Market Principles

The Governance-Logic should enable a real-time environment for software that is market orientated. Software should be able to acquire supply side security by booking resources in advance, pay a premium for on demand service or qualify for discounts by waiting on standby. These market models make resource consumption more effective and efficient as software can utilise the cycles of low demand to process work where time is not critical, for example, during an automated routine backup. The Governance-Logic should present Mobile Agents with market events through the Mobile Agent services interface within the Governance-Layer-Interfaces of Layer-3. These market events may for example indicate when a price for a resource has passed a threshold, allowing software to behave intelligently as it responds to its environment. For example, when the price of computational process or bandwidth reaches a certain point, software may sell back a portion of some of its resource contract at a higher price than it originally paid.

Software should perceive costs in much the way a business would. Prices for computational resource should fluctuate on the 3i-Node, depending on demand from other local software. As the price for resource climbs, software can outsource non-core tasks to components located on adjacent or nearby 3i-Nodes. Ultimately, a long-term price hike could see the software move permanently to a more cost effective location.

It is the Governance-Logic that enables the real-time property of the 3i-Node as it provides the ability to do computation on demand as a function of money. An example application is a real-time phone call, where fixed bandwidth is required for the duration of the call.

2.3.1.3. Governance-Layer-Interfaces

The Governance-Logic should service the Governance-Layer-Interfaces, which are the only means information at Layer-4 has of communicating with the outside world. It is

[5] See III-Node Currency, Promissory-Money, 3.1.1.2

here that Mobile Agent can purchase resource contracts or communicate with Mobile Agents on other 3i-Nodes. Governance-Logic should process these information requests, enforce the legislative rules determined by the 3i-Custodianship and translate them into the correct action at the Interoperability-Layer-Interface. The Interoperability-Layer should then take these governance directives and implement them directly on the Layer-1 Host-Platform.

2.3.1.4. Legislation

Government-Logic should implement Automated-Legislation and Directive-Legislation by utilising the Governance-Layer-Interface to exchange information with Mobile Agents. However, independent 3i-Custodianship processes should implement passive legislation, such as in tracing money[6] or maintaining databases of Promissory-Money[7].

[6] See Money-Tracing, Track-And-Kill, 3.5.6

[7] See III-Node Currency, Promissory-Money, 3.1.1.2

2.3.2. Reference-Membrane

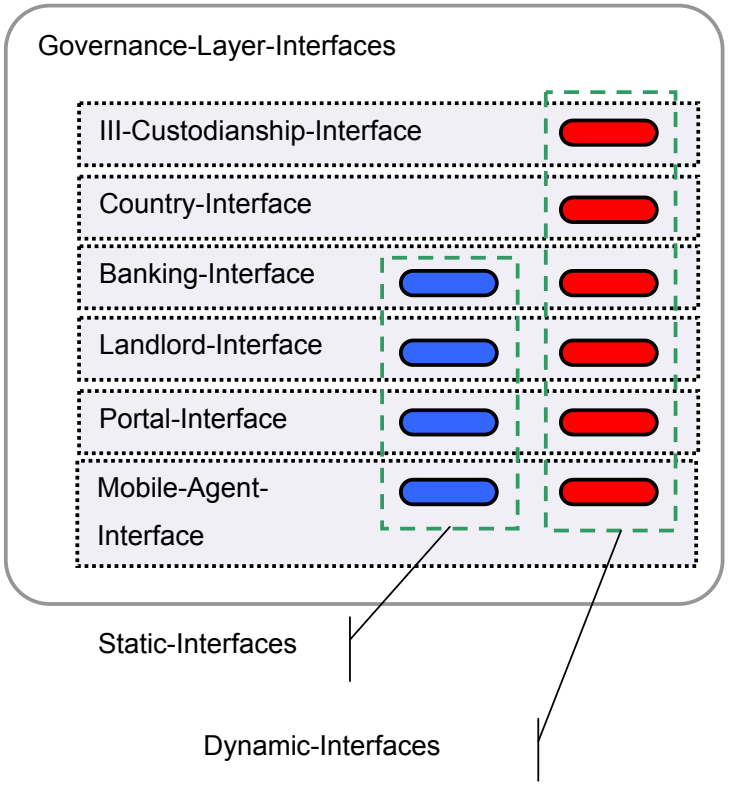
The Reference-Membrane was conceptualised by Dr. Fielden and is an extension of the isolation mechanism that preserves Layer-3 & Layer-4 integrity. The Reference-Membrane should act as a filter to strip out all Active-Content and references from communication between the Governance-Layer-Interfaces and the Governance-Logic. The Reference-Membrane is critical in preventing the transferring of references between Layer-3 and Layer-4, which could otherwise lead to the corruption of the 3i-Node and allow mal-ware to propagate.

The Reference-Membrane prevents unknown software from establishing references within the Governance-Layer, even though information is exchanged. Specifically, it should do this by firstly ensuring that it does not actively exchange an internal reference directly during inbound communication. Secondly, it should prevent any active process from being exchanged during outbound communication, which may interact in a manner undetermined at design-time with components in the Governance-Layer. Allowing active process as a return parameter could enable the exchange of a Governance-Logic

reference to the active process, which would simply forward the reference to the unknown software at Layer-4.

Thus, the inability of software to interfere with the infrastructure machinery of the 3i-Node - the universe in which it operates - prevents it from attaining privileges associated with the governing superstructure. This means software can be handled safely, and obviates destructive processes including viruses, worms, Trojan-horses, spy ware, Spam and hacking. Spy-ware is ineffective, as its perspective is limited to its own boundaries. In the case of a Trojan horse, even if a masquerade of usefulness is successful, it too is unable to manipulate anything beyond its own confines.

Governance-Layer-Interfaces



The Governance-Layer-Interfaces should connect information at Layer-4 to the outside world via the Governance-Logic facilitator, although the interfaces themselves should do no explicit work.

In order to preserve forwards and backwards compatibility of software on the 3i, some interfaces should retain their context forever by using non-deprecating Static-Interfaces. These interfaces interact with generic software and should include the Banking-Interface, Landlord-Interface, Portal-Interface and the Digital-Business-Interface. The 3i-Custodianship-Interface and the Country-Interface should interact with software designed to manage ongoing legislative and infrastructure processes by governments, and so should be expected to change. All the interfaces should have a Dynamic-Interface that should be available to offer additional functionality as determined by the 3i-Custodianship.

2.3.3. 3i-Custodianship-Interface

The 3i-Custodianship-Interface should manage the Governance-Layer superstructure and include the managed upgrade strategy that prevents a Super-Crash scenario. The interface should provide the highest level of control over the 3i and be managed by the 3i-Custodianship. It is through the 3i-Custodianship-Interface that new Governance-Logic should be loaded.

The 3i-Custodianship-Interface should also be responsible for implementing global directives such as Information-Banning that could be applied for pirated music or movies; or Information-Tracing for assisting global interests such as combating terrorism.

2.3.4. Country-Interface

The Country-Interface should be a centralised point of control for individual governments to configure their national digital information policy. The interface should allow governments to enforce Portal-Search-Warrants, Portal-Confiscations, monitor

phone calls, trace banking transactions and set rates of taxation. It should also enable countries to register the arrival of foreigners into the country, so that their Portals may adopt the national digital legislation. The Country-Interface should utilise a Dynamic-Interface as people and not software engage it. It should therefore change in accordance to that specified by the 3i-Custodianship.

2.3.5. Banking-Interface

The Banking-Interface should be the only way software can convert the mobile form of money known as Promissory-Money into local currency known as Realtime-Money and back again. This interface should allow the 3i-Custodianship to trace Money-Transfers or Cluster-Transfers when it is reformed back onto a other recognised currencies on the 3i called Brokering Agent currency. A single Money-Transfer of $10 million US is equivalent to a billion transactions of one cent each performed during a Cluster-Transfer.

The Banking-Interface should also be responsible for the registration of all Embedded-Charging transactions. The Governance-Logic should trace the path taken by all Promissory-Money, to ensure that all Embedded-Charging transactions have been registered, so that the appropriate tax can be paid[8].

The Banking-Interface should record all the information required to form an automated accountancy process. It should include the endpoints of the network where traditional currency may be exchanged for Promissory-Money via banking clearinghouse software.

2.3.6. Landlord-Interface

The Landlord-Interface should provide a mechanism whereby the owner of the Layer-1 Host-Platform can configure 3i-Node specific parameters. These include: -

- Setting prices for Layer-1 resources.

[8] See Chapter 3 – Monetary Architecture

- Configuring partner details on the peer-to-peer network, schedule shutdowns and identify itself on the 3i.
- Setting the amount of Promissory-Money in circulation[9].
- Acquiring other currencies in exchanging for Promissory-Money.
- Ability to authorise as the genuine owner of the 3i-Node.

2.3.7. Portal-Interface

The Portal-Interface should enable software to assist people in managing their information. The Portal-Interface should allow Mobile Agents to communicate directly with people[10].

2.3.8. Digital-Business-Interface

The Digital-Business-Interface should allow Mobile Agents to engage the host 3i-Node with activity such as entering into resource contracts (see Mobile Agent Communication in the next section). Mobile Agents should access the following functionality from the Layer-3 Governance-Layer: -

- Assign a fixed bandwidth communication channel directly with another Mobile Agents on a partner 3i-Node.
- Purchase a fixed rate of computational process for a fixed time.
- Purchase additional computational process on demand at a premium price.
- Set price thresholds that notify the Mobile Agents when computational process is at a price that makes it worthwhile to sell back the contracted resource.
- Replicate Mobile Agents to adjacent 3i-Nodes.
- It may receive notification that it has been involved in a Recursive-Cluster-Trace with the 3i-Custodianship and has been suspended.

[9] See Currency Valuation, Issue-Level, 3.1.2.6

[10] See User Communication, 2.5

2.4. Network Services

The 3i infrastructure is more than the collection of 3i-Nodes. It should also include competitive information services driven by the open market.

2.4.1. Routing-Agents

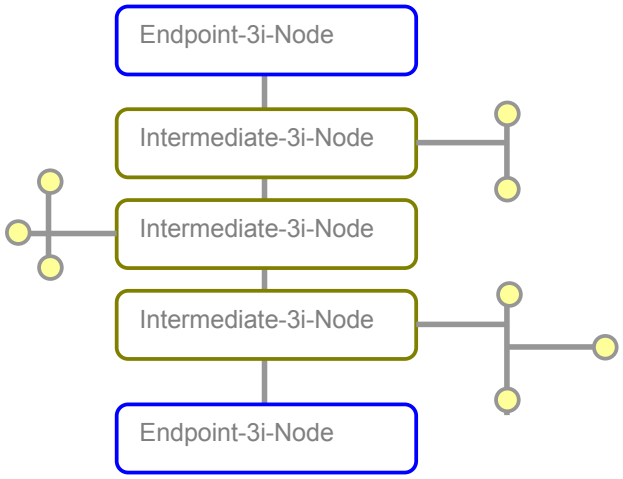
Routing-Agents are Mobile Agents that provide absolute peer-to-peer path information to other Mobile Agents.

2.4.1.1. Peer-to-peer Routes

These databases are dynamic in that as new partnerships may be configured on the 3i, the routing database is likely to be updated. For example, during a telephone call between two Portals, there must exist an absolute peer-to-peer path between the 3i-Nodes that host the Portal Mobile Agents. In the event of failure of one of the intermediate 3i-Nodes in the peer-to-peer chain that connects the Portals, then a new route may be sought. It is likely that the Mobile Agents managing this process would be prepared to pay a premium for immediate and substantial computational process in order to minimise the downtime during the route switch.

Alternatively, the Mobile Agents may request multiple routes pre-emptively to further minimise this time. Ultimately, in mission critical applications, the Mobile Agents may decide to broadcast in parallel across multiple routes.

2.4.1.2. Path Selection



The client Mobile Agents is likely to provide only one of the Endpoint-3i-Nodes, which represents its current location.

Determining the destination Endpoint-3i-Node may be determined by any number of the following: -

- 3i-Node identification, where the Mobile Agents is required to be, such as during a logon through a public Portal.
- Cost effective resource pricing for rental of space or computational process.
- Level of trust[11]
- Reliability-Rating[12]
- Security-Rating[13]
- Minimum load level[14]

[11] See Currency Valuation, 3i-Node Integrity, 3.1.2.1

[12] See Currency Valuation, Reliability-Rating, 3.1.2.3

[13] See Currency Valuation, Security-Rating, 3.1.2.5

[14] The guaranteed minimum load a Digital-Business may schedule.

2.4.2. Brokering-Agents

Brokering-Agents are Mobile Agents that generate revenue for their owners through the trading of Promissory-Money issued by a multitude of 3i-Nodes. Brokering-Agents can issue their own Promissory-Money, which is expected to be more stable than 3i-Node promissory money because: -

1. Brokering-Agent Promissory-Money represents an average of all the currencies that it holds.
2. Brokering-Agents are not obliged to establish intrinsic value by issuing Realtime-Money, whose value is expected to fluctuate in response to the demand for computational resource on the host 3i-Node.

Brokering-Agents may further stabilise their currencies by holding other established Brokering-Agent currencies. Ultimately, Brokering-Agents that invest heavily in other stable Brokering-Agents become super stable points of reference during currency conversions[15]. This reduction of risk during

[15] See The Monetary-Platform, Clearinghouses, 3.7

currency exchange means traders can reduce the margin between buy and sell rates, which can lead to more efficient clearinghouses.

2.4.3. Reference Emulation

A Digital-Business should only be capable of accessing objects that it has references to on the Layer-2 Interoperability-Layer. It should not be possible to access references on other 3i-Nodes directly nor references held by other Mobile Agents on the same 3i-Node. However, since this is a fundamental requirement for computing, references on remote 3i-Nodes should be emulated at Layer-4. All locking and synchronisation should be handled by a Mobile Agent, which may utilise Embedded-Charging for its services.

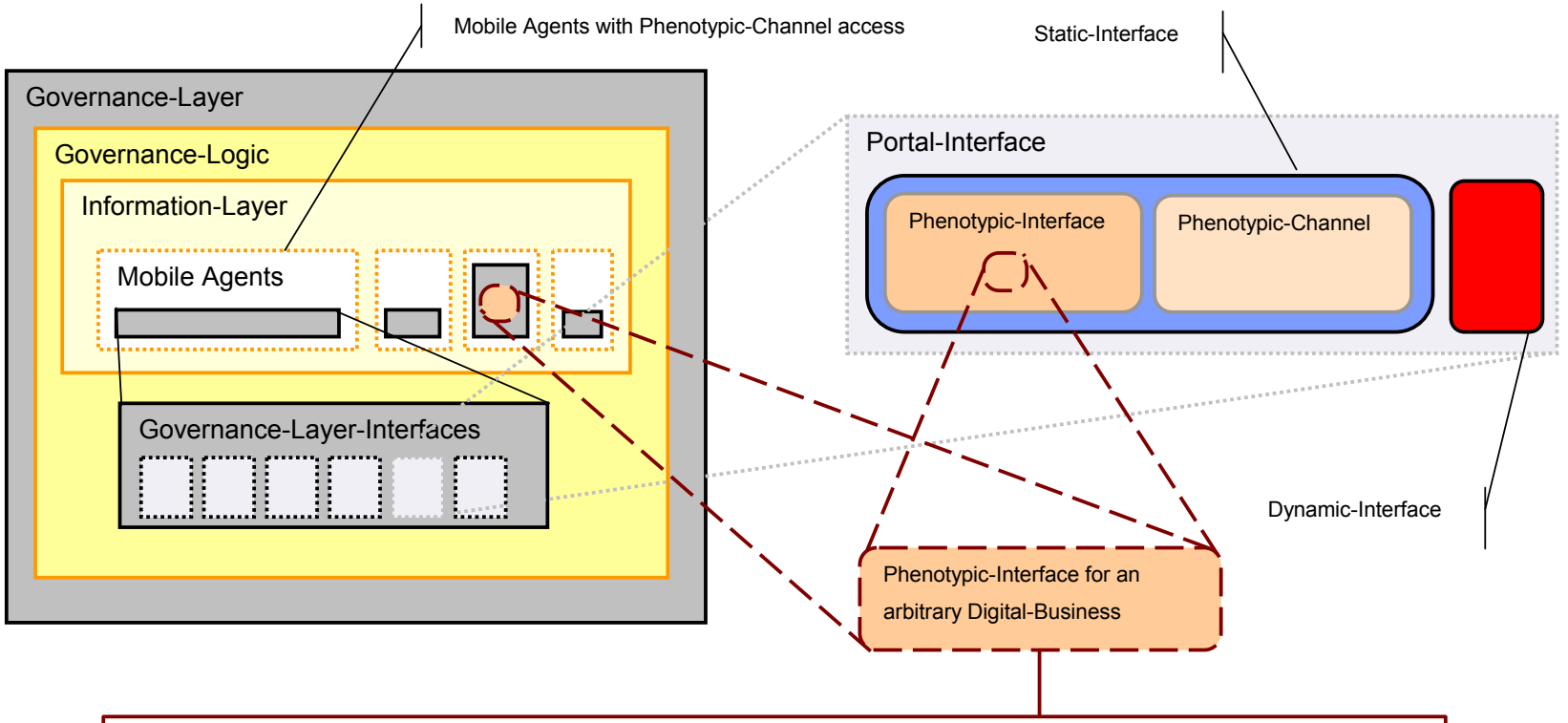
2.4.4. Landlord-Agents

Landlord-Agents are Mobile Agents that utilise the Landlord-Interface to configure the 3i-Node with respect to its peer-to-peer partners. The Mobile Agents should be able to configure bandwidth allocation and set pricing for resources[16]. For example, a mobile phone may have access to a wide bandwidth channel. It is critical that the Landlord-Agent correctly reflect the telecommunication charges for this channel so that the device owner does not lose money. The Landlord-Agent may adjust pricing dynamically, for example, if the battery is running low, it may increase the price of computational process.

2.5. User Communication

All Mobile Agents should have the capacity to communicate directly with the user as a sensory experience when the opportunity arises. However, only authorised Mobile Agents, known as Portals, should be able to directly access the information expression equipment by authorisation process known as a Phenotypic-Channel. Although the Portal has privileged communication ability to access the display device and speaker, it does so through the same Phenotypic-Interface that the ordinary Mobile Agent should use.

[16] For a complete list of functionality see Landlord-Interface, 2.3.6



The Phenotypic-Interface should have the property of being able to embed information that conforms to the Phenotypic-Interface seamlessly. NOTE: This diagram has been simplified for conceptual purposes. This reference is actually an Escrow-Reference (see Escrow-References in Digital-Business Communication in the next section).

2.5.1. Phenotypic-Interface

The Phenotypic-Interface should be an object that enables a Mobile Agents to express scene information as if it were a virtual-reality experience.

2.5.1.1. Specification Development

The Phenotypic-Interface specification should be defined over a long period of time and the specification may never be complete. However, this inevitable incompleteness will not affect forwards and backwards compatibility. Initially, the most important sensory information is likely to be defined, such as vision and sound. As more is learned about the nature of the human senses, other sensory information can be added to the specification. Adding resolution to the specification will not prevent forwards compatibility, because software written in the future can still operate on previous Layer-2 implementations, with the only affect being that the old implementation is unable to interpret modern sense information. So for example, an old version may not be able to recreate the sense of smell, but is likely to be able to reproduce sound.

2.5.1.2. Omnipotent-Translation

The Phenotypic-Interface should allow for Omnipotent-Translation, which allows for multiple Mobile Agents describing themselves as a sensory experience to occupy a single experience without destroying the context or ability for a person to perceive that experience. Omnipotent-Translation should only accept Escrow-References[17]. Escrow-References prevents the Portal from actually gaining access to the objects that it represents.

This sensory embedding should be a function of the Layer-2 Interoperability-Layer, which converts a sensory experience of a single Digital-Business to the Phenotypic-Interface into 3-Dimensional spaces by rendering the visual experience into an enclosed sphere. Other sensory information should be approximated in relation to the size and location of each sphere to the user's virtual location. For example, objects that are small or far away should have their volume levels proportionally muted. The sensory attenuation of the

[17] See User Communication, Digital-Business Communication, 2.5.4

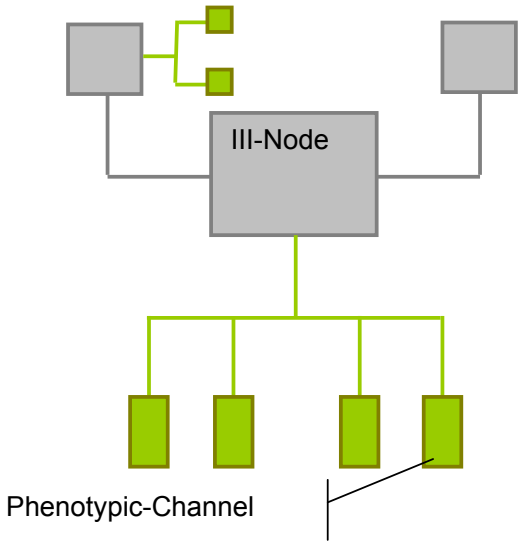
enclosed object that has undergone Omnipotent-Translation should be defined in the Phenotypic-Interface-Specification.

2.5.1.3. Transcendence

The Portal should allow the user to switch from an omnipotent view of all the objects to a sensory view of an individual object. The omnipotent view should include all objects that have each undergone Omnipotent-Translation and are arranged arbitrarily by the user within their Portal. The sensory view of an object should be defined by the Phenotypic-Interface. The object should be rendered directly as if the Mobile Agent had authorised the Phenotypic-Channel itself.

The switching is known as Transcendence and should occur through an interactive cue from the user when they indicate a closer examination of an object. The handover should be beyond the awareness of the Mobile Agent, and should be regained by the Portal once the user indicates their wish to disengage via a Portal specific signal.

2.5.2. Phenotypic-Channel Authorisation



Each 3i-Node should be capable of implementing any number of Phenotypic-Interfaces. Mobile Agents should be able to acquire any one of these Phenotypic-Channels.

2.5.2.1. Keyed-Authorisation

Keyed-Authorisation should occur when a Mobile Agent wishes to take control of a Phenotypic-Channel. The Mobile

Agent presents the Identification number of the Phenotypic-Channel and the cryptographic key to establish integrity. These components should be configured through the Landlord-Interface, most likely through dedicated software that assists the 3i-Node in their configuration.

2.5.2.2. Handover-Authorisation

Handover Authorisation should occur when a reference to a Phenotypic-Interface that has been authorised as a Phenotypic-Channel is passed to another Mobile Agent. Handover-Authorisation is expected to occur when a logon application identifies and authorises a new Portal. The logon application, which currently utilises the Phenotypic-Channel to conduct the authorisation process, should hand the reference over to the new Portal.

2.5.3. Portal-Legislation

Since Portals manage user information and communicates directly with the user, they are in a privileged position and should implement various forms of legislation issued by the 3i-Custodianship to protect the user.

2.5.3.1. Locality

The 3i-Custodianship should be able to acquire the logical location of a Portal to assist law enforcement. The Governance-Logic should be able to match the Portal logon session with the 3i-Node, so that the physical location of adjacent 3i-Nodes can be used to assist in finding the user.

2.5.3.2. Security Education

Users should acquire sufficient knowledge of the security model in order to conduct safe transactions. The security model should be simplified to a single decision for the user. The user should be required to understand that giving an object to another object enables the receiver to have complete access of everything contained within the object[18].

The 3i-Custodianship should require Portals to educate users to ensure that they understand the implications of transferring references before they be allowed to operate the Portal. This mechanism may be implemented through a

[18] See Interoperability-Layer, Introspective-Scope, 2.2.2

simple explanation and test that user performs on initiating a new Portal.

Users should be encouraged to access information via their own Portal and share individual objects and not simply share an entire Portal. Since users may access their Portals from any other Portal, there is no reason to utilise another user's Portal. Allowing other users to utilise a Portal is undesirable because additional users can jeopardise the integrity of the Portal by transferring references to malicious objects. Additional users are likely to inadvertently alter the arrangement of objects in a Portal during their Portal activities. The original user is unlikely to be aware of these alterations, and it may make it more difficult to find objects that are arranged through action orientated sorting mechanisms, such as the object history. It is likely to make law enforcement easier if users are responsible for activities conducted from within their own Portal.

2.5.3.3. User Protection

In addition to educating the user, the Portal should provide additional protection where possible for the transferring of key references. This may protect users from mistakes or through coercion from malware. Although these references should still be transferable, they should require suitable warnings that notify the user of the implications of their actions. These references would normally include entire collections of objects, such as a history list, work area or a list of friends. A legitimate action for transferring references could be when a user intends to migrate to a new Portal, but wishes to retain their personalised information.

2.5.3.4. Interface-Shielding

The 3i-Custodianship should define a dataflow that governs all information access for applications within a Portal. Portals should be legislated against the introduction of additional interfaces that prevent information moving freely to another Portal. The Phenotypic-Interface should be all that is required to express information, which preserves forwards and backwards compatibility of all software.

2.5.4. Mobile Agent Communication

2.5.4.1. Pre-emptive-Interface

Mobile Agents should be able to present multiple alternatives that pre-empt its next state. These should be presented as a list of objects in the same format as the Phenotypic-Interface-Specification. The Pre-emptive-Interface should then notify the object of which selection the user took.

Although not vital to the 3i objectives, the Pre-emptive-Interface is likely to create state awareness across all objects that can be used to reduce complexity and increase productivity. Users can view the progression of each object over time. This may be as broad as stepping back in time to view the state of the Portal in the previous day, or as narrow as operating on a single object. For example, the user may utilise a paragraph in a document that existed a few days earlier, while the actual document that contained the text was deleted only hours ago.

2.5.4.2. Resolution-Identifier

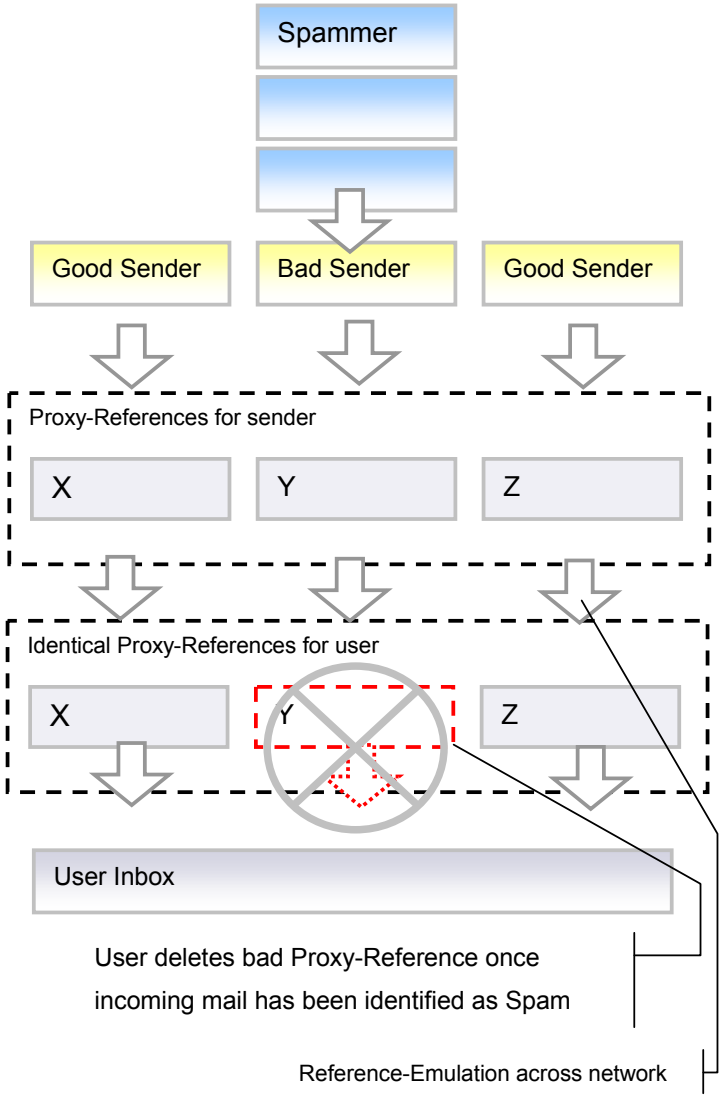
The Resolution-Identifier should provide the Mobile Agent with an opportunity to simplify its communication in relation to its level of user awareness. The Resolution-Identifier should provide information to the Mobile Agent as to how easily the user is able to ascertain detail within the Phenotypic-Interface. This should be measured as a percentage. A value of 100% should indicate a complete virtual reality experience through Transcendence[19], while a number that approaches 0% should indicate a higher level of attenuation as the Mobile Agent competes for awareness with other information in an omnipotent view.

In some situations where the communication device with the user is limited, such as with a small screen size on a handheld computer, the Resolution-Identifier may never reach 100% - even during Transcendence.

[19] See Phenotypic-Interface, Transcendence, 2.5.1.3

2.5.4.3. Spam Prevention

Spam should be prevented by user communication through private references. Communication software should release a unique reference for each user of inbound information. These users should deliver their mail through these Proxy-References. In the event that a sender of mail violates the integrity of their Proxy-Reference by forwarding it to an entity that invokes it as a Spam address, then the user should simply delete the Proxy-Reference without affecting the private relationships it has with other users. In the event that the Integrity violation of the Proxy-Reference was an accident, the user can issue the sender with a new Proxy-Reference. Reference-Emulation enables references to work across 3i-Nodes.



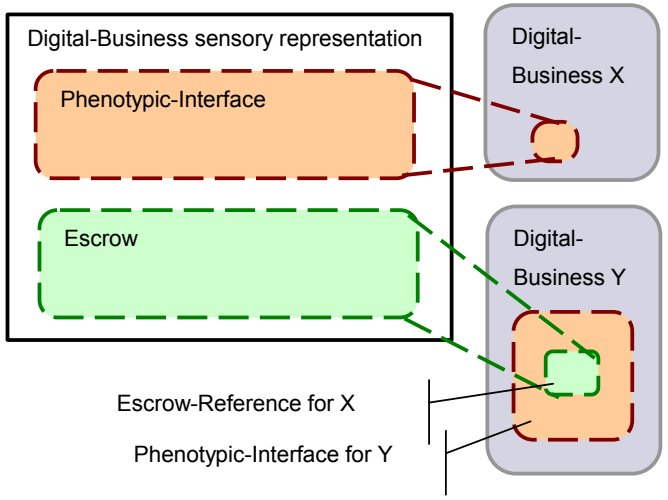
2.5.4.4. Private-Reference-Lookup-Interface

Since user communication should be performed through private Proxy-References, it should not be possible for users to publish these references directly in a public space. Doing so would result in spamming processes from attaining these references and users would constantly be deleting bad Proxy-References.

2.5.4.5. Escrow-References

Escrow-References provide Mobile Agents the opportunity of maintaining a reference to an object without actually being able to access the object itself. This allows Mobile Agents to specialise as information organisers, without the user of the information having to trust the Integrity of the software. This prevents a Trojan-horse from masquerading as a useful and safe product, which actually acts as spy-ware on the objects that it manages.

This mechanism may also be managed at a much lower level in Set Theory by linking two Sets and hiding the contents from each other.



2.5.4.6. Protect Sensitive Information

Mobile Agents should be able to detect the Integrity of the user log in, so that sensitive information can be protected. For example, if a Portal is accessed from within a bank's own physical security system and using a retina scanner, the Integrity can be assumed to be high. Specialised security objects that may hold sensitive information such as money could then unlock themselves. The 3i-Node should make this information available through a registration process with the 3i-Custodianship or other trusted party who would issue 3i-Nodes with a security rating.

Users should be encouraged to protect sensitive information, as the weakest link of the 3i (like all software systems) is the human operator.

3. The Monetary Architecture

The 3i should have the fundamental property of a distributed monetary platform, whereby all value, whether a physical resource or intellectual property can be traded in an automated fashion.

3.1. 3i-Nodes

The 3i should be composed of a multitude of 3i-Nodes, interconnected to form a peer-to-peer network. Each 3i-Node should be able to link to any number of partners. Mobile Agents should be free to move through the network, paying their way for bandwidth or computational resource on each 3i-Node.

3.1.1. 3i-Node Currency

An unbounded and open grid network would soon be prone to parasitic behaviour if Mobile Agents were not forced to pay for the resource that they use.

To enable a financial structure, each 3i-Node should be responsible for the management of a virtual currency that has parallels with a monetary system of a country. The currency should have a value because it is backed on the computational resources of the 3i-Node to perform useful work.

3.1.1.1. Realtime-Money

Realtime-Money should be stored in the Mobile Agent bank account, which should be maintained by the Governance-Logic. The Realtime-Money currency should be unique for each 3i-Node and recognised as a legal tender for Mobile Agents occupying the 3i-Node. Each payment made by the Mobile Agent for resources utilised should be deducted from this account. Realtime-Money is an artificial phenomenon and is the result of an architectural requirement that enables money to be exchanged in real-time, and hence requires a minimal amount of computational resource to enable trade.

The Governance-Logic should ensure that this account is always in credit, and destroy insolvent Mobile Agents[20].

[20] See Governance-Layer, Governance-Logic, 2.3.1

3.1.1.2. Promissory-Money

Promissory-Money should be stored as a collection of extremely large numbers, which may be exchanged for Realtime-Money on presentation to the 3i-Node that issued it through the 3i-Custodianship-Money-Exchange[21]. Since Promissory-Money should only be presented once, it is likely to be presented immediately by the new owner during an exchange transaction. Failure to do so means the new owner runs the risk that a previous owner may attempt to claim the equivalent Realtime-Money that it represents. This is likely to be a standard process for any monetary exchange. Once presented, the Promissory-Money should be converted into Realtime-Money for the new owner who is likely to convert it in a second transaction back into its flexible Promissory-Money form. Since Promissory-Money is portable it can be encrypted and moved off the 3i-Node, so it is likely to be the financial mode of choice for Mobile Agents.

Although 3i-Nodes issue Promissory-Money, which is the act of issuing legal tender, it still does not constitute as credit[22]. The sale value of Promissory-Money is not equivalent to its redeemable value on the 3i-Node. This apparent lack of equivalence is actually the 3i-Node purchasing the entire value of its Promissory-Money in advance in a market paradigm that embeds a perceived level of risk for the 3i-Node's ability to honour its currency[23].

The issuing of Promissory-Money should require 3i-Custodianship intervention and tracing to prevent the 3i-Monetary-Platform from evolving into a facilitator of anonymous payments[24].

3.1.2. Currency Valuation

Brokering-Agents are Mobile Agents that establish rates of exchange between multiple 3i-Node currencies. The

[21] See The Monetary Architecture, Money-Flow, 3.2

[22] See Monetary Platform Stability, No Credit, 3.4.2

[23] See Currency Valuation, III-Node Integrity, 3.1.2.1

[24] See The Monetary Architecture, Money-Tracing, 3.5

Promissory-Money issued by the 3i-Node is only as valuable as a Brokering-Agent is willing to back it. Ironically, it is important that the 3i-Custodianship allow 3i-Nodes to cheat and make a profit where possible, as this creates monetary stability[25].

The following factors are expected to play a role in this evaluation: -

3.1.2.1. 3i-Node Integrity

Brokering-Agents should be profit centred, and so use trust orientated rules derived from Game Theory[26]. These ensure that in the event a 3i-Node tactically devalues its currency suddenly, the Brokering-Agent that owns its Promissory-Money will be exposed to a limited loss, reducing the incentive for a 3i-Node to cheat. Newly established 3i-Nodes are likely to experience this effect by receiving a poor rate of exchange when they first establish themselves on the 3i network – even though the work they perform is really of a much higher value. Over time, they may end up paying for the entire value of their issued Promissory-Money.

The currency trading risk is carried by Brokering-Agents. When a new 3i-Node is formed, a Brokering-Agent is likely to set a very low trust rating for the new currency, which translates into a low exchange rate for the new 3i-Node. As the 3i-Node consistently performs at a level higher than its currency valuation, Brokering-Agents are likely to raise its valuation as more Mobile Agents begin to desire the currency.

3.1.2.2. Work Capacity

Brokering Agents should determine rates of exchange of 3i-Node Promissory-Money. Once the 3i-Node has established its trust, the rate should match the value of the underlying 3i-Node resource. Mobile Agents translates Promissory-Money into Realtime-Money and spend it back at the 3i-Node where it was originally issued. Therefore, all things being equal, a 3i-Node that is capable of delivering strong utility is likely to have a stronger currency.

[25] See Monetary Platform Stability, Legalising Currency Corruption, 3.4.1

[26] Popularised by John von Neumann

3i-Node utility should be founded on the following: -

- Computational process
- Electrical power
- Disk
- Bandwidth between partner 3i-Nodes.

3.1.2.3. Reliability-Rating

Mobile Agents are likely to utilise Routing-Agents to determine suitable 3i-Nodes. One of these elements is the 3i-Node's Reliability-Rating. An extremely high rating indicates an impeccable uptime, making it suitable for latency sensitive applications like telephone calls. A low rating may be assigned to a PC that intermittently connects to the 3i through a dialup connection.

Software that utilises low rated 3i-Nodes is likely to have to operate processes in parallel for redundancy. Low rated 3i-Nodes therefore command a lower utility price than highly rated 3i-Nodes. This effect is likely to taper off, as some software may still find value in an unreliable service. An example of this could be a software agent responsible for providing a distributed backup. Although some of the backup locations could be housed on reliable but more expensive 3i-Nodes, a larger number could be housed on cheap unreliable 3i-Nodes. The philosophy here is that in the event of failure of the reliable 3i-Nodes, data could ultimately be recovered when the first unreliable 3i-Node comes back online.

3i-Nodes can increase their rating by providing early warning of impending shutdowns, which give Mobile Agents time to replicate themselves to other 3i-Nodes or reconfigure themselves. A connectivity failure with a partner that does not provide suitable shutdown notice will have a negative impact on the 3i-Node's Reliability-Rating and the potential revenue it can earn, so it is in the interest of the high rated 3i-Node to choose its partners carefully.

3.1.2.4. Locality

3i-Nodes should compete with each other in an open market[27]. Prices can be expected to fluctuate according to the principles of supply and demand. 3i-Nodes that are positioned well are likely to command a premium on their otherwise equal utility in the same way that Content Distribution Networks (CDNs) do over TCP/IP.

3.1.2.5. Security-Rating

Routing-Agents are likely to provide Security-Rating information to Mobile Agents. The Security-Rating is a risk assessment of the likelihood that a 3i-Node may be a malicious host. Poorly rated sites, however, can still be utilised by sensitive processes, but Mobile Agents are self limited in the functionality that they can undertake on the 3i-Node. Mobile Agents are likely to keep sensitive information, such as Promissory-Money-Codes, encrypted and only utilise the 3i-Node for transport or storage and not for general processing. 3i-Nodes with a high Security-Rating are likely to command a premium on identical utility, compared to low-rated sites.

3.1.2.6. Issue-Level

3i-Nodes can devalue their current exchange rate in relation to other currencies by issuing more Promissory-Money into circulation. This can to be done by reducing the currency price relative to other currencies[28]. Although this may increase 3i-Node revenue in the short-term, the long-term effect is likely to undermine the value of the currency. 3i-Nodes should therefore only issue as much Promissory-Money as is required in the network for efficient handling and holding purposes.

3.1.2.7. Issue-Stability

Brokering-Agents should be able to issue Promissory-Money too. This currency is anticipated to attain a value that averages all the currencies it holds. Brokering-Agents are likely to reinvest in other Brokering-Agents to further stabilise

[27] See Governance-Logic, Market Principles, 2.3.1.2

[28] See The Monetary-Architecture, Money-Flow, 3.2

their currency, which is likely to act as a currency-strengthening factor.

3i-Nodes are expected to be more exposed to currency fluctuations, because the value they can deliver is tightly coupled with the price that they offer for their utility. It is the price of this computational resource that is likely to fluctuate on a 3i-Node as the market forces of supply and demand surge with Mobile Agent use. Price fixing is an unlikely alternative, as once capacity is saturated; non-performance for Mobile Agents simply decreases the Reliability-Rating.

3.1.2.8. Hybrid Evaluations

Some 3i-Nodes may attempt to stabilise their currencies by ensuring it is backed by currencies from Brokering-Agents. The higher the currency valuation the 3i-Node can attain, the better price it obtains for its utility. Currency stability should be managed during 3i-Node capacity upgrades. 3i-Nodes that increase their utility over time should offset this with an increased issue of Promissory-Money, locking their valuation in relation to their chosen Brokering-Agents.

3.1.3. Establishing a 3i-Node

Engaging a 3i-Node with the 3i should be a seamless process automated by software and should require minimal human intervention.

3.1.3.1. Starting a 3i-Node

In starting a new 3i-Node, the owner of the 3i-Node should establish a Layer-2 Interoperability-Layer implementation and install the Layer-3 Governance-Layer that has been transcoded to the specific 3i Candidate implementation. The owner should find suitable partners who are already connected to the 3i. Partners can be found either through a public co-ordination process, such as may be required during a peer-to-peer network of mobile phones; or it can be done privately, as may be arranged by telecommunication companies as they interconnect massive data channels. Completing the connection requires the configuration of the Layer-1 Host-Platform implementation to enable the bandwidth connection to the chosen partners. Finally, a Landlord-Agent should be invoked to configure 3i-Node. A Landlord-Agent is a Mobile Agent that utilises the Landlord-

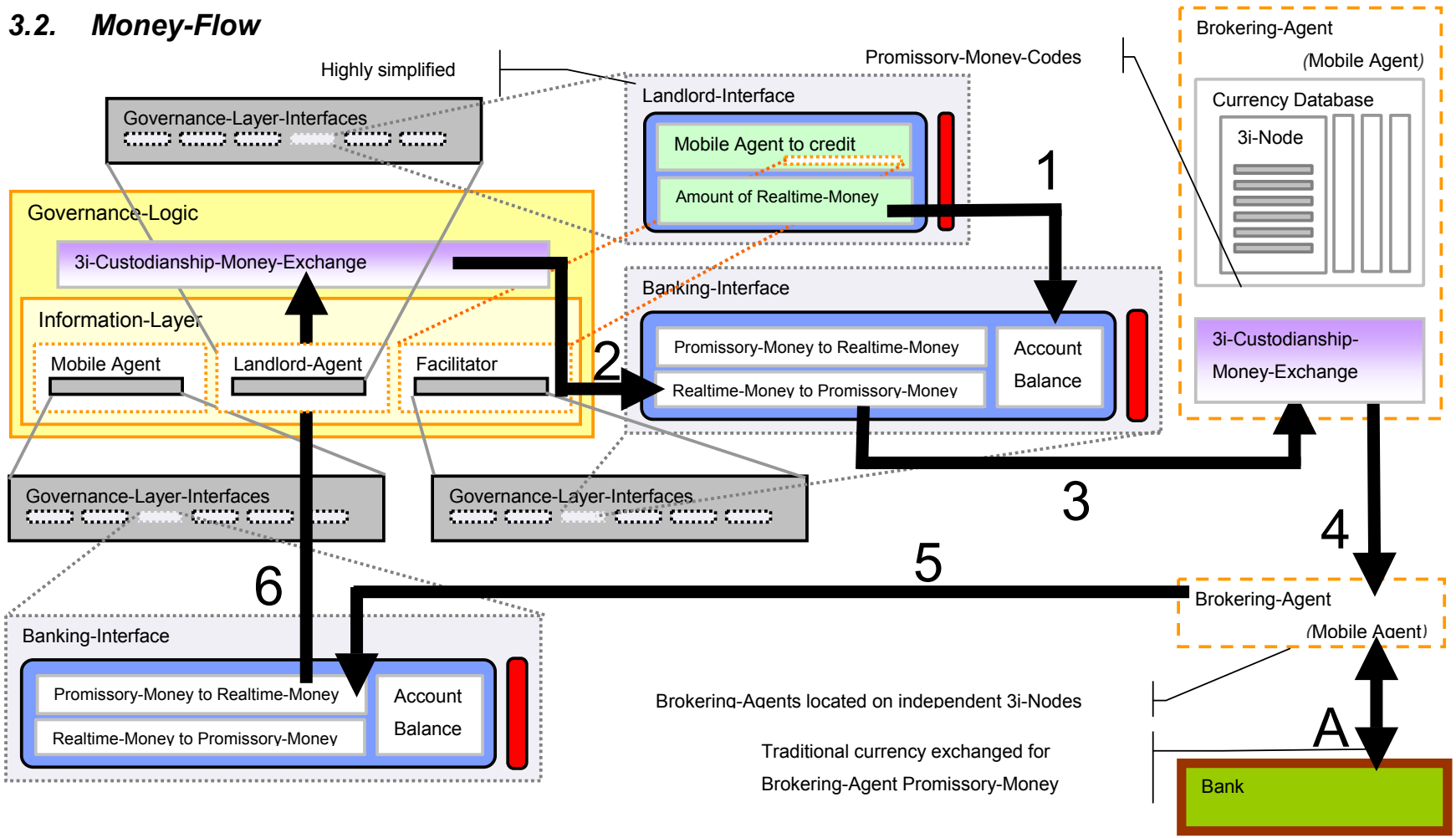
Interface to set up the logical bandwidth connection and register the new 3i-Node with the 3i-Custodianship. 3i-Custodianship should acquire sufficient details of the logical location of the 3i-Node in relation to other transport protocols in the broader GII so that it may assist law enforcement in tracking the user's location.

3.1.3.2. 3i-Node-Inheritance

3i-Node-Inheritance is likely to occur when owners of existing 3i-Nodes perform reconstruction both physically or logically. A physical reconstruction could occur if the Layer-1 Host-Platform was destroyed. A logical reconstruction could occur if the owner wished to establish new partners in a logically different location in the peer-to-peer network.

3i-Node-Inheritance is important because the 3i-Node does not lose its identity, and still honours the issued Promissory-Money. This is advantageous, because the 3i-Node is likely to obtain better exchange rates with Brokering-Agents by using its already established currency, rather than starting a new 3i-Node.

3.2. *Money-Flow*



3.2.1. Flow Key

1. A Digital-Business that has authorised access to the Landlord-Interface assigns Realtime-Money directly to itself or into a facilitating Mobile Agents 3i-Node account.
2. The facilitating Mobile Agent converts Realtime-Money into Promissory-Money by using the 3i-Custodianship-Money-Exchange, which tracks issued money and the Identification number of the Mobile Agent that owns it.
3. The Promissory-Money is transferred to an arbitrary Brokering-Agent, again using the 3i-Custodianship-Money-Exchange.
4. The Brokering-Agent exchanges Promissory-Money with other Brokering-Agents and swaps information as to where Promissory-Money can be found for each currency. This data contains the location of the issuing 3i-Node and the list of Brokering-Agents that hold this currency.
5. A Mobile Agentwhose account balance is low and requires Realtime-Money to avoid delinquency, exchanges Promissory-Money from sources other than its local 3i-Node with a Brokering-Agent in exchange for Promissory-Money of its local 3i-Node.
6. The Mobile Agent exchanges the local Promissory-Money for Realtime-Money. Governance-Logic updates the Promissory-Money information by destroying the Promissory-Money-Codes.

A. Banks may exchange traditional currencies, such as the US Dollar with Brokering-Agents that have established suitable trust through a stable exchange rate. These currency conversion are required to ultimately move money in or out of the 3i-Monetary-Platform.

3.2.2. Trade

Trade is expected to occur when Promissory-Money changes ownership, and most likely there is either an exchange in the real world of goods, service, information or other money in parallel; or a digital exchange inside the 3i has taken place through Embedded-Charging. The exchange of Promissory-Money should be managed completely by Mobile Agents.

During a transaction, each Promissory-Money-Code (a large number that comprises Promissory-Money) should be presented to the issuing 3i-Node and exchanged for Realtime-Money – then, in a second transaction, it should be translated back into Promissory-Money, but this time with a different series of Promissory-Money-Codes that are now unknown to the original owner.

3.2.3. Issue-Level Fluctuations

The total amount of money in circulation for a 3i-Node is equal to the sum of all the Realtime-Money accounts for each Mobile Agent, plus the total amount of issued Promissory-Money. During the Money-Flow cycle, the total amount of money in circulation can fluctuate: -

1. When a Mobile Agent spends Realtime-Money at a 3i-Node for local resources, the total amount of money in circulation decreases.
2. When a 3i-Node issues Promissory-Money to a Digital-Business, there is no change of money in circulation.
3. When a Landlord-Agent assigns Realtime-Money to a facilitator Mobile Agent, the total amount of money in circulation increases.

3.2.4. Convergence of Promissory-Money and Traditional Currency

3i-Node currencies should have intrinsic value as they are backed by productive computational work. Traditional currencies have value because there is trust that these currencies will not suddenly devalue, either because they are partially backed by precious metals or a credible government that uses sensible fiscal policy issues them.

Ultimately, a computationally backed currency system may become more stable than those backed by the good intentions of government[29]. This possibility exists because collectively Promissory-Money is fully backed and the underlying value does not change. Although the price of hardware falls logarithmically, the stability of the 3i-Monetary-Platform is most

[29] See The Monetary Architecture, Clearinghouses, 3.7

likely to remain unaffected. Landlord-Agents are expected to increase the Issue-Level to match the increased output capacity. The 3i-Node will be capable of doing more work, so it is in the interest of the 3i-Node take corrective action to demonstrate a stable 3i-Node currency.

3.3. Booting the 3i-Monetary-Platform

The 3i-Monetary-Platform should be initiated with a fixed and controlled number of services that form a crude money exchange known as the Intermediate-Money-Cycle. Once stabilised, the money platform can be migrated to a completely free market. Initiating the 3i-Monetary-Platform should require careful planning, because the forces of supply and demand during the Intermediate-Money-Cycle can initially only be expected to operate effectively within a reasonable range of prices. Specifically, when the price for computational process is too low, the Intermediate-Money-Cycle may no longer be capable of generating sufficient revenue to maintain the trading services required to maintain the process. For example, if computation process fetches a low price, it is unlikely to be able to fund a service such as a prepaid music download.

3.3.1. Intermediate-Money-Cycle

To reduce the risk of dysfunction, it is suggested that an Intermediate-Money-Cycle be initially achieved, whereby the 3i peer-to-peer network is first established. It is possible that Mobile Agent functionality can operate normally without a monetary platform, the only risk being than parasitic agents will begin to utilise the resource because it is free.

In a second stage Intermediate-Money-Cycle, the 3i-Nodes begin charging for resource and encourage the Mobile Agents to pay -- those that are not upgraded will have limited access to resource or occasionally be destroyed if they use too much resource because they are already technically bankrupt.

The Intermediate-Money-Cycle attempts to prevent the value of Promissory-Money from crashing relative to ordinary currency. This may mean that the 3i-Monetary-Platform should be manually adjusted by altering the Money-Balance and the Utility-Balance. Under normal market conditions, both these component are interrelated and self-regulating.

3.3.1.1. Money-Balance

The Intermediate-Money-Cycle should strive to maintain Money-Balance, by ensuring that there is sufficient inbound money supply into the system to match outbound money demand. The early 3i architects should control the inbound money supply by providing a managed number of applications that utilise the computational resources of the 3i (in exchange for ordinary currency like USD). Outbound money is likely to be applications that require purchases with ordinary currency. The challenge during the Intermediate-Money-Cycle is to manually balance both the supply and demand for currency until such time that the network has sufficient mass to support open services, whose balance can be maintained by market forces.

3.3.1.2. Utility-Balance

The Intermediate-Money-Cycle should maintain Utility-Balance by ensuring that there is sufficient network utility to meet demand. Consumer applications of computational processing power, storage and bandwidth should match the output of the sum of 3i-Nodes.

3.3.2. Migration to full platform

The Intermediate-Money-Cycle should be utilised until there is sufficient activity that free market and public services can be added without risking the business viability of other services already on the network by crashing the 3i currency. For example, if a successful free market service such as a prepaid software licence service[30] becomes too successful too early, a massive oversupply of computational process may upset the Utility-Balance. The oversupply would be a result of users selling their processing power in order to pay for their software license. The oversupply of computational process would reduce its price and degrade the integrity of the Intermediate-Money-Cycle. Not only would the system be unable to fund the software licenses, but every service that relied on computational process to fund it would suffer.

It is important that the Intermediate-Money-Cycle reach a suitable mass where additional applications can be added

[30] A prepaid software license service may utilise computational process on the local machine to generate revenue. This revenue is used in a second transaction to pay for the licenses of software using a rental model.

without the risk of upsetting either the Utility-Balance or Money-Balance. Once this point has been reached, the Intermediate-Money-Cycle function can be switched from growing the network to simply maintaining stability of the network. Once the manual management of the 3i currency has been marginalised, it may be discontinued from use.

3.4. 3i-Monetary-Platform Stability

The 3i-Monetary-Platform is likely to be stable for the following reasons: -

3.4.1. Legalising Currency Corruption

The 3i-Custodianship should allow 3i-Nodes or Brokering-Agents to rapidly devaluing their currency. This devaluation is most likely to violate the currency's integrity with Brokering-Agents. This counter-intuitive liberty is expected to stabilise the network as Brokering-Agents are forced to incorporate the probability that 3i-Nodes will devalue their currency if it profits them to do so. 3i-Nodes are likely to be required by Brokering-Agents to back their own currencies by performing work at below market rates (as per Game Theory). Collectively, this forms an Evolutionary Stable Strategy, as occurs during the evolution of life.

Using legislation to prevent 3i-Nodes from devaluing their currencies only creates a false sense of stability. At any point, if 3i-Nodes are able to devalue on mass during an economic attack, the Brokering-Agents would be forced to devalue their holding currencies to stay solvent – an act that is likely to create distrust and instability in the 3i-Monetary-Platform.

3.4.2. No Credit

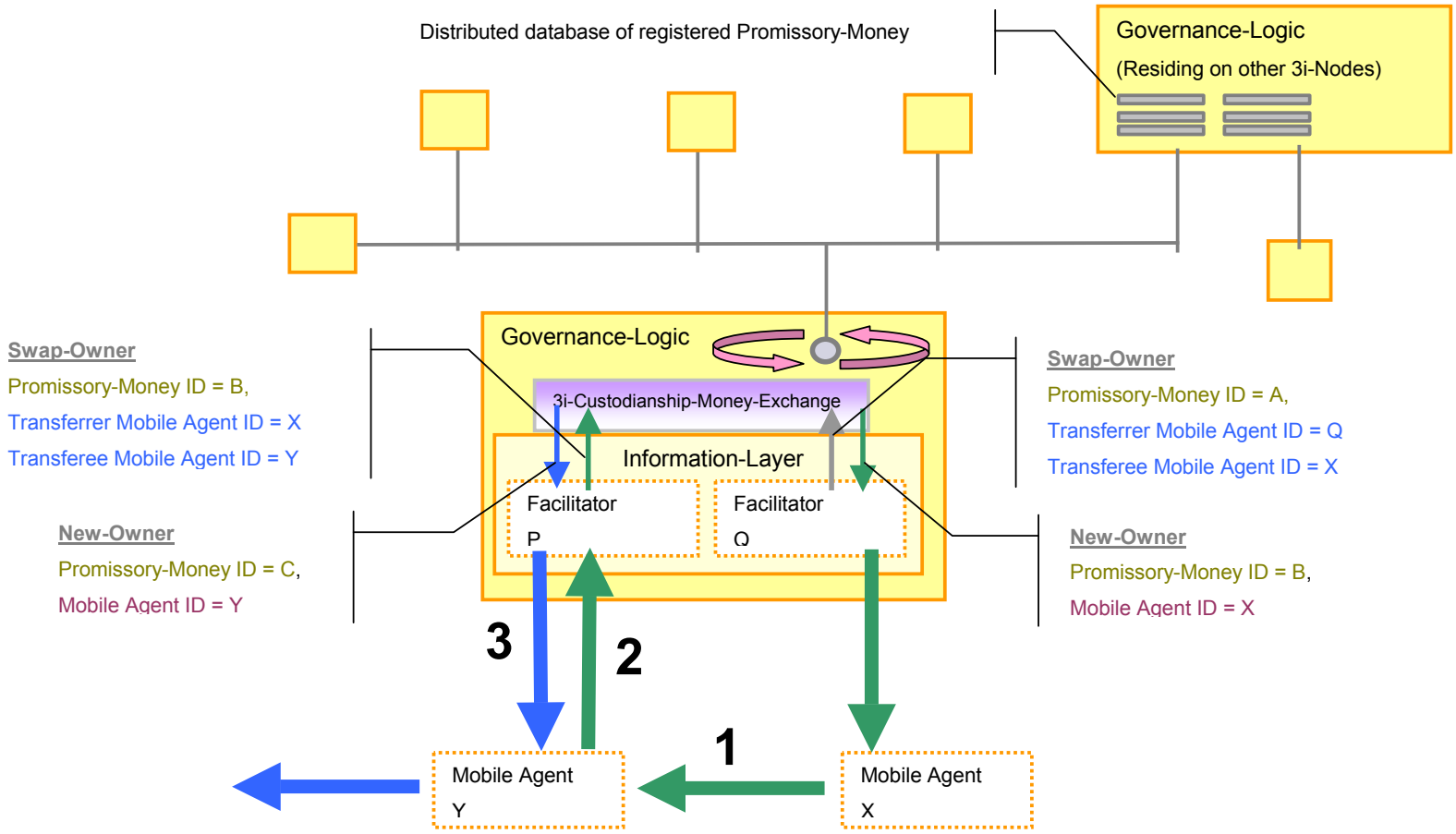
The Governance-Logic of a 3i-Node should not give credit to Mobile Agents, regardless of their authentication or positive identification as a trusted source. Credit is unnecessary, and reliance on it only forms a point of artificial stability that may collapse in the future. Such as scenario where legions Digital-Business clones suddenly refuse to honour their debts may causing a spike in the Money-Balance.

Governance-Logic should not require 3i-Nodes to maintain stability of their currency, as Brokering-Agents should provide this incentive by improving the rate of exchange for stable 3i-Node currencies.

3.4.3. Distributed

The 3i-Monetary-Platform should be distributed and be able to absorb multiple simultaneous failures.

3.5. *Money-Tracing*



3.5.1. Exchanging Promissory-Money

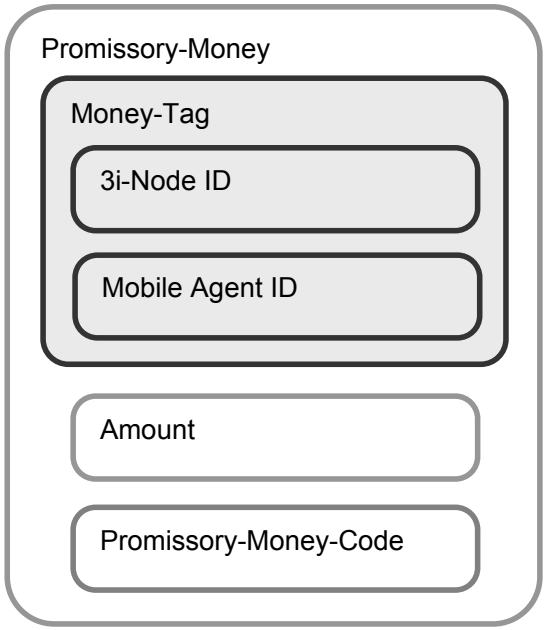
Promissory-Money is mobile and so is expected to form the foundation for the exchange of money.

The process for exchanging Promissory-Money is outlined in the above steps: -

1. A Mobile Agent forwards Promissory-Money to another Mobile Agent.
2. The receiving Mobile Agent converts the Promissory-Money into Realtime-Money. In the process it should declare whether it is an Embedded-Charging transaction or a Money-Transfer.
3. The Governance-Layer returns a new Promissory-Money object registered to the new Mobile Agent.

3.5.2. Money-Tags

Money-Tags should be a foundation tool that enables the tracing of money.



Mobile Agent are likely to formally exchange Promissory-Money (Step 2) through the 3i-Custodianship-Money-Exchange. The 3i-Node should only honour Promissory-Money-Codes that are registered to a listed Mobile Agent ID.

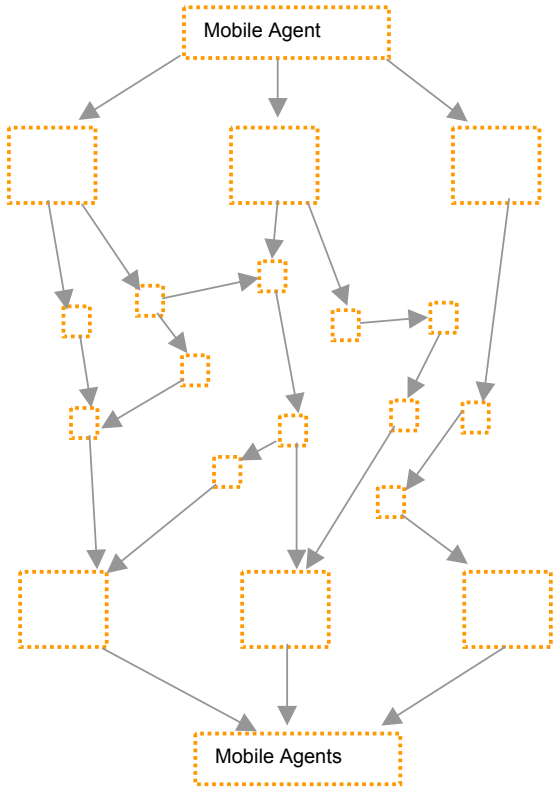
The 3i-Custodianship should be responsible for issuing Mobile Agent IDs as part of the Governance-Layer function of creating a new Mobile Agent.

3.5.3. 3i-Custodianship-Money-Exchange

The 3i-Custodianship-Money-Exchange is a multitude of Mobile Agents managed by the 3i-Custodianship that provide a conduit for other Mobile Agents to commit Money-Transfers into the GII to connect conventional money with 3i currency. These Mobile Agents should maintain a database of Money-Tags accessible to the 3i-Custodianship to enable money tracing.

3.5.4. Money-Fragmentation

A core function for global stability is the ability to track funds. Doing so in the 3i becomes more complex because Mobile Agents can use any number of facilitating Mobile Agents in conjunction with a large number of smaller transactions. This environment makes it unfeasible for people to attempt to track individual transactions.



It is not possible to track individual transactions, as money is fluid and can easily be fragmented and reconstituted into units of almost identical value. An intermediate Mobile Agent attempting to evade tracking is unlikely to forward an identical amount to another intermediate, as this would be represent a simplistic transaction that is easy to detect. Instead, transfers paths are likely to be scrambled, borrowing technologies from fractals to encryption to ensure paths and amounts are effectively random.

3.5.5. Money-Laundering

In order to prevent Money-Laundering, it should not be possible for a Mobile Agents to send money to another without its consent. By empowering Mobile Agents to actively determine which other Mobile Agents they are authorised to receive funds from makes them accountable for those funds.

Mobile Agents should be responsible for all funds that they move. In the event that they move laundered funds, they may face destruction while the government under which the author is registered may reclaim all the Promissory-Money-Codes that they hold. The only exception should be during an Embedded-Charging transaction. This public transaction creates natural inefficiencies for transferring money, as each transaction is reduced by at least the minimum tax imposed by the 3i-Custodianship. Mobile Agents should not be punished for receiving funds from a legal Embedded-Charging transaction. It is therefore possible, but severely limiting to launder money in using the 3i by scrambling fragmented transactions with Embedded-Charging. However, at any stage it should be possible for the 3i-Custodianship to manually examine money exchanged through Embedded-Charging transactions through the 3i-Custodianship-Money-Exchange.

3.5.6. Track-And-Kill

The Track-And-Kill process should be used to prevent the movement of illegal money through the 3i. The following sequence outlines the Track-And-Kill process: -

3.5.6.1. Lockdown

Lockdown is the identification of a Mobile Agent involved in the generation or movement of illegal money. The Mobile Agent receiving money from one or more Mobile Agents should be locked down.

3.5.6.1.1. External-Lockdown

Laundered money may originate on the outside of the 3i, within the GII. Money generated through illegal activity may be channelled through a bank and into a Mobile Agent. Banking authorities or companies operating as exchanges between the GII and 3i should identify the Mobile Agent to which the transactions have been made.

3.5.6.1.2. Internal-Lockdown

The following situations may result in a Mobile Agent undergoing Internal-Lockdown: -

- A Mobile Agent may be identified as subverting Embedded-Charging control for globally banned content.
- A Mobile Agent may have utilised Embedded-Charging to obtain some shielding by paying tax on what should be an illegal Money-Transfer.
- A Mobile Agent may commit an illegal Money-Transfer.

3.5.6.2. Recursive-Cluster-Trace

Software should be used to identify all Mobile Agents involved in transactions with the Mobile Agent that has been locked down. The 3i-Custodianship software is likely to automatically detect zones where money may be transferred into a broader 3i as legal Money-Transfers. These Mobile

Agents should be eliminated from the search, still many innocent Mobile Agents may be included in the cluster[31].

3.5.6.3. Cluster-Sanity-Check

The Recursive-Cluster-Trace is likely to utilise probability rules to make decisions on whether to include a Mobile Agent in the cluster. For example, Mobile Agents that have simply replicated themselves to perform fragmentation can easily be identified. An 3i-Custodianship technician should do a Cluster-Sanity-Check to ensure the total amount of money approximately matches that lost in the GII. The cluster may be refined by manually sampling Individual Mobile Agent. Mobile Agents are likely to fall into three distinct zones. These zones each include Mobile Agents that either: -

- Are guilty and are solely involved in the fragmented money transaction
- Are guilty and deliberately attempt to implicate innocent Mobile Agents through sophisticated coercion that cons them into accepting a Money-Transfer of a small amount instead of an Embedded-Charging transaction.
- Are innocent and are being deliberately implicated into an illegal transaction.

Manually sampling Mobile Agents at the periphery of zones is likely to reduce the number of innocent Mobile Agents involved the cull. Mobile Agents can reduce this risk of accidental termination by taking care not to accept unauthorised funds.

3.5.6.4. Engage Kill

In the final stage, all Mobile Agents involved in the Recursive-Cluster-Trace are suspended. All their funds are frozen, in whatever 3i-Node or Brokering-Agent currency it may reside. All Promissory-Money should be exchanged through the 3i-Custodianship-Money-Exchange; so frozen money can be identified within the 3i-Custodianship database. The Mobile Agent should receive notification of its own suspension through an independent Mobile Agent that is not frozen so that a legal owner may attempt to reclaim

[31] See Entity Digital-Rights, Digital-Businesses, **Error! Reference source not found.**

important information if required. During this time, the 3i-Node should support all rental activity, but prevent any process usage. If no application for retrieval has been placed within a specified time, the Mobile Agent should be destroyed.

3.6. Embedded-Charging

Any Mobile Agent should be able to engage in trade with another Mobile Agent via Embedded Charging (EC) or transfer money. Both of these functions should occur through interfaces managed by the 3i-Custodianship via each 3i-Node's Governance Layer. During an EC transaction, the Mobile Agent is expected to declare the income (via the EC Interface) and pay tax on the revenue generated. Failing to do so by avoiding trade through the EC Interface should be considered as tax evasion and is likely to expose the Mobile Agent to Lockdown during a Recursive-Cluster-Trace. Governments are expected to introduce sophisticated software tools to spot patterns in Money-Transfers that may be taxable.

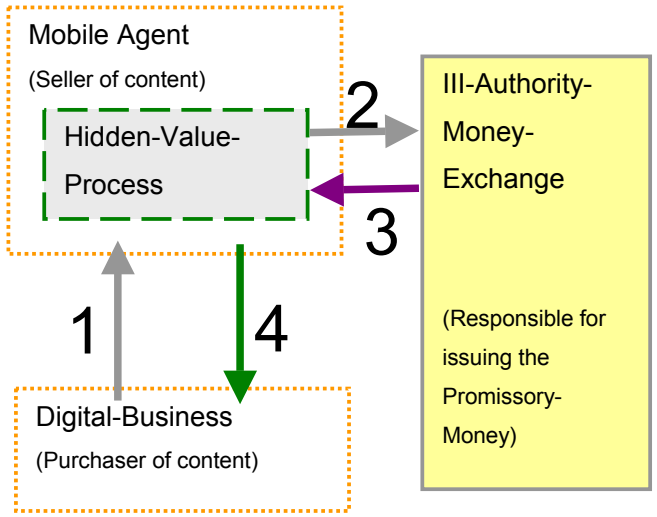
3.6.1. Taxation

As a value creation principle, tax should be charged on revenue, regardless of expenses.

Tax = TAX_RATE x Revenue

Total economic activity should be viewed as finite. Mobile Agent should be rewarded when revenue is increased.

3.6.2. Active-Content



Active-Content is information that is interactive and contains process. By hiding functional components of the Active-Content within Mobile Agents in a Hidden-Value-Process, and only allowing the outputs of the Mobile Agent to be included in an Embedded-Charging transaction, the Active-Content can be protected from reverse engineering.

In the steps above: -

1. The Mobile Agent that intends purchasing the content is likely to pay the Seller Mobile Agent with Promissory-Money.
2. The content selling Mobile Agent is expected to exchange the purchaser's Promissory-Money with the 3i-Custodianship-Money-Exchange, residing on a 3i-Node or on a Brokering-Agent. It is expected that the exchange be declared as an Embedded-Charging transaction that is taxable revenue.
3. The Seller should be issued with updated Promissory-Money by the 3iCustodianship-Money-Exchange.
4. The content selling Mobile Agent is likely to release the desired output from the Hidden-Value-Process.

3.7. Clearinghouses

The 3i-Monetary-Platform is expected to be more efficient at clearing funds than banks. Both banks and Brokering-Agents exchange money by agreeing on buy and sell exchange rates that best reflect the value of the relative currencies. The following factors are expected to increase the level between the buy and sell rates for clearinghouses and decrease it for Brokering-Agents.

3.7.1. Currency Fluctuations

Over many transactions, currency trading by banks becomes inefficient because currencies may change in value. This is not so much at the point when the transaction occurs, but rather that the owner of the clearinghouse must increase the margin between the buy and sell rates to cover losses incurred in the event that a holding currency devalues. Although, statistically, banks may expect to break even during these apparently random cycles, it may often not do so, as environment changes that may lead to a change in exchange rates are likely to be perceived by traders prior to the exchange rate adjustment.

The 3i-Monetary-Platform is not exposed to the complexities of trading risk across the same geopolitical landscape that banks have to manage. For example, it may only take a government policy introduction, such as exchange control legislation, to radically affect the value of the currency. The 3i-Monetary-Platform is one of the most abstract forms of money possible, backed entirely on identical units of processing power. It is likely to be more stable than traditional currency, as the promise to pay may be distributed trillions of times across 3i-Nodes and Brokering-Agents, all playing the same rules of trust epitomised by Game Theory.

Unlike traditional currency, the promise to pay does not rest with government fiscal policy. The mass of virtual currency is expected to form a stable homogenous entity that may provide stable rates between currencies. For example, if the Euro were to devalue, it is unlikely to affect the exchange rates of other virtual currencies that may hold stable Brokering-Agent Promissory-Money.

3.7.2. Human Effort

Clearinghouses are required to cover costs not incurred by Brokering-Agents. These traditional business costs include rent, employees and human error. The portion of this is incrementally smaller for large transaction volumes. However, the 3i-Monetary-Platform requires limited human effort. It requires software for the Brokering-Agent trust assessment algorithms and it needs computing resources to execute the brokering.

3.7.3. Competition

Brokering-Agents are expected to be prolific, as they should not be limited from exchanging money provided they utilise the 3i-Custodianship-Money-Exchange. This high level of competition is expected to reduce excess profit taking from Brokering-Agents.