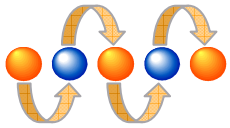


Information Supply Chains

Transcending The Glass Ceiling of Web Services to a Value Chain of Information

- Prototype Phase 1 Report -

Who?

Who am I?

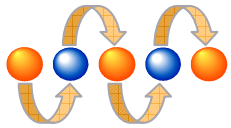
Christopher Ball

Technical Strategist

Christopher.Ball@CGIFederal.com

Who are we?

An Open Source Open Standards based collaboration



Striving to create Value Chains of Information

UBMATRIX™

IPHIX

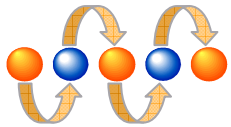
CGI

IONA

Akira

What are we creating?

Virtual Team of 25+



Information Supply Chain

Open Source Prototype

Whitepaper

Who are you?

- Agency . . ?
- Role . . ?

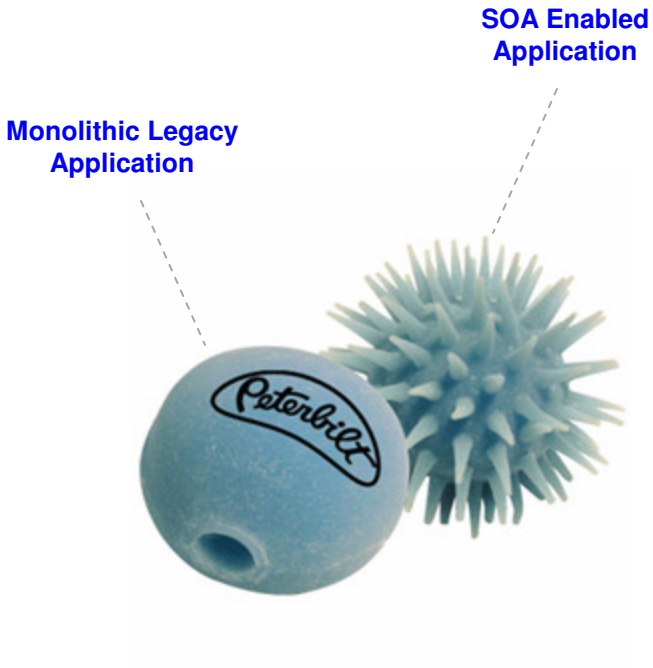
Table of Contents

CGI

SOA

Turning Applications Inside Out

- SOA is the evolution of **integration** . . .
  - From integration as a **tactical solution**
  - To integration as a **fundamental practice** and **core competency** of an enterprise



Consequences?

"**SOA is Analogous to Splitting an Atom -** The concept is remarkably simple, yet how to go about doing so and the consequences of having done so, are exceedingly complex."

- Christopher Ball

"Service-oriented architectures are like snowflakes . . . no two are alike"

- David Linthicum

The Good

Enhanced Value Proposition . . .

EAI

SOA

Value Facet	Value	Metric	Some Practical Manifestation
Utilization	To get a higher return from existing Information Technology resources	Degree of use	• Utilizing existing Data • Utilizing existing Functionality
Visibility	To easily look into the enterprise, see its shape, understand its dynamics, and comprehend and measure its behavior	Ease of access	• Ease of access to desired Data • Ease of access to desired Functionality • Ease of access to desired Process
Agility	To easily answer tomorrow's questions and implement tomorrow's ideas with today's technology	Ease of change	• Ease of assimilating new or differing types of Data • Ease of assimilating new or differing types of Functionality • Ease of changing Process
Specialization	To use external resources for that which is not core to the value the business adds and to externally offer that which the business does best	Use of what is best	• Ease of discretely accessing differing types of Data • Ease of discretely defining process participation by Data or Functionality
Resiliency	The ease with which today's solutions can be maintained in tomorrows technology and approaches	Ease of maintaining	• Ease of assimilating new or differing types of Data • Ease of assimilating new or differing types of Functionality • Ease of changing Process
Community	The ease with which a collection of entities with a common interest can collaborate and work in a choreographed fashion	Ease of collaborating	• Broad reuse of information and real-time access to a broad set of rich information
Effectiveness	The ease with which an organization achieved its objectives.	Ease and accuracy in achieving objectives	• The having the right information available to make the right decision at the right time.

CGI

Qualities of SOA

Role	Description	Facet
Qualities	Degree to which a particular condition or state is exhibited	Abstract
		Loosely Coupled
		Runtime Bound
		Flexible
		Reusable
		Scalable
		Extensible
		Incremental
		Refinable
		Highly Cohesive

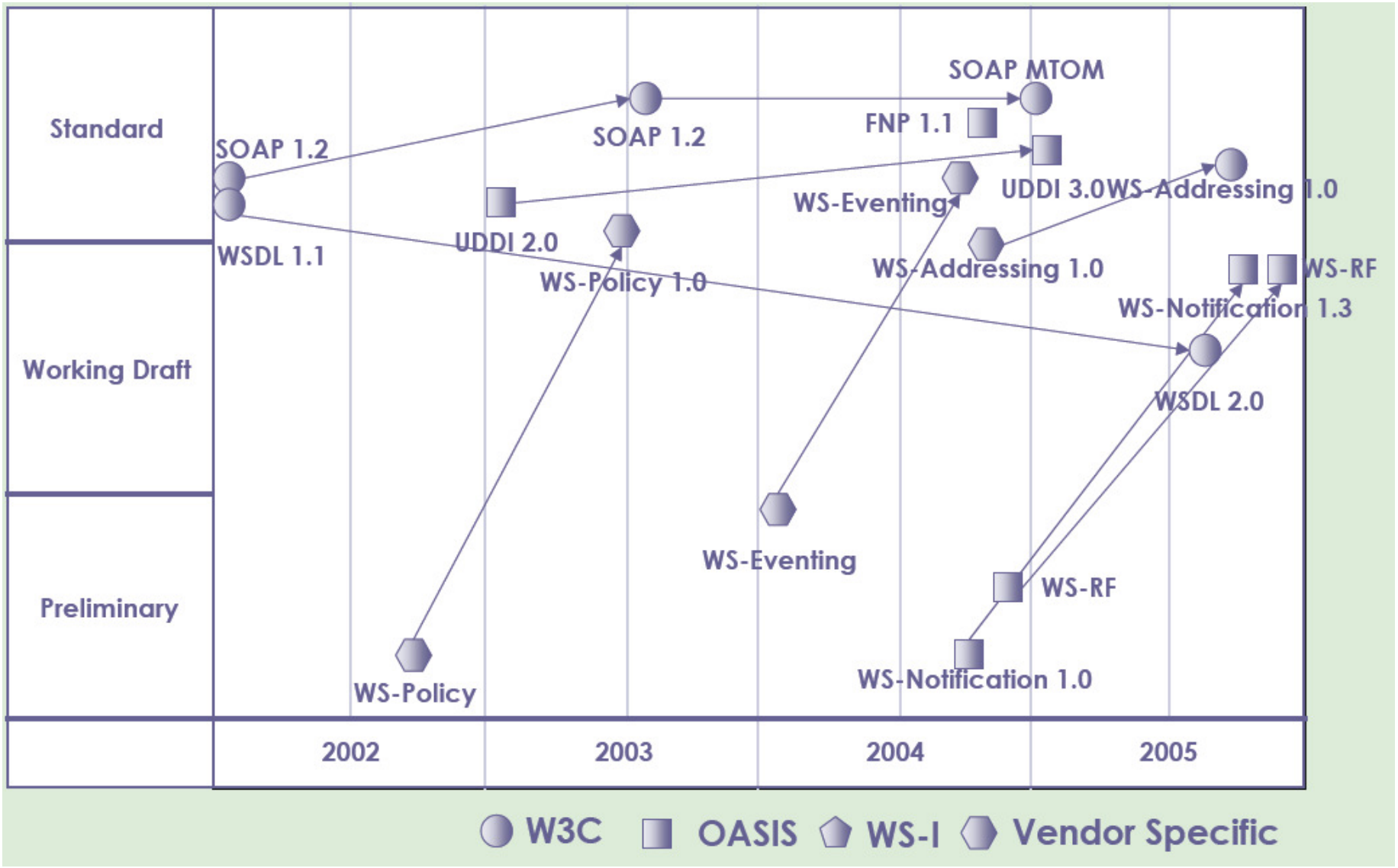
CGI

Fundamental and Best Practices . . .

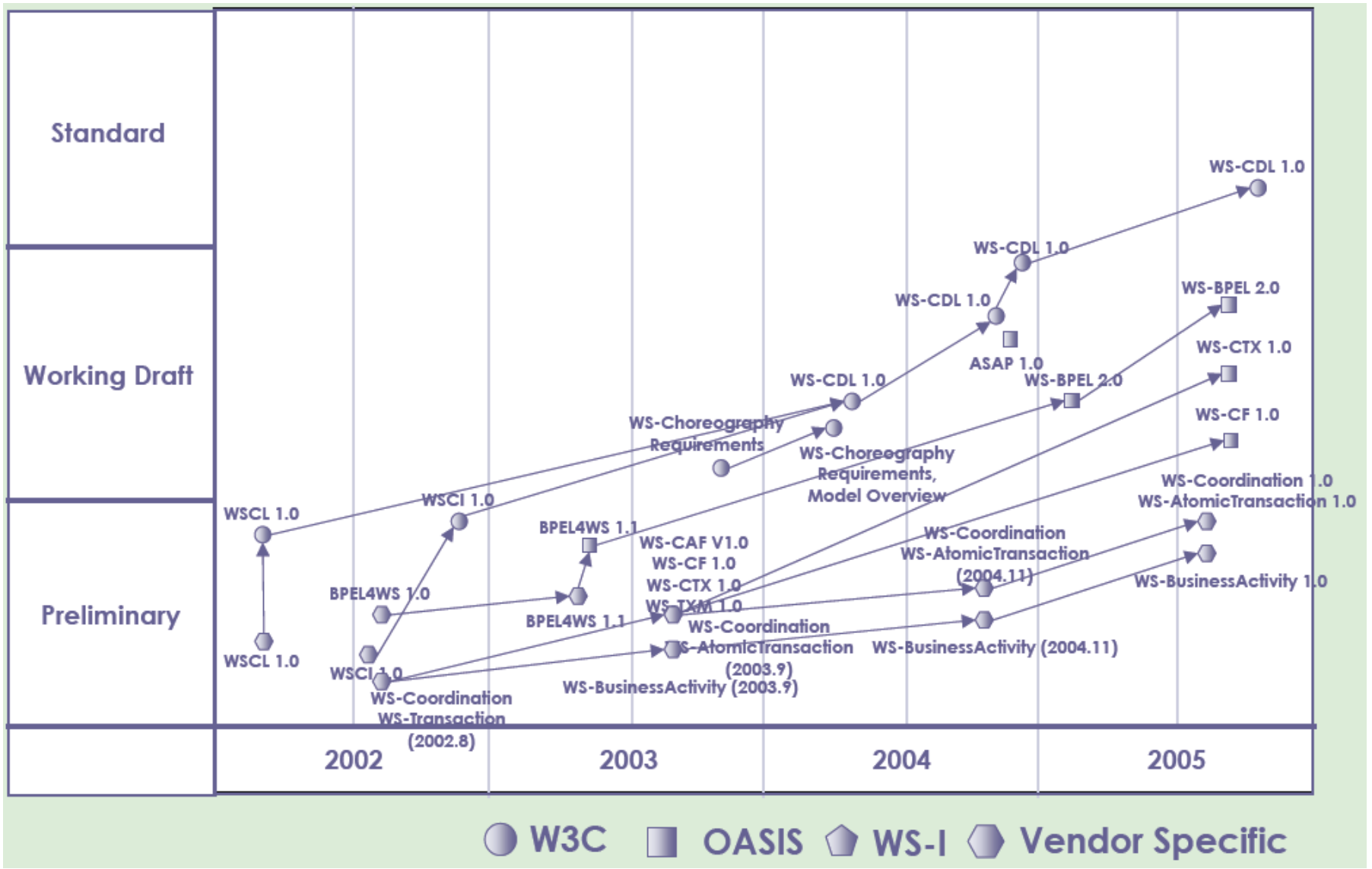
Role	Description	Facet
Fundamental Practices	Practices which are indispensable in pursuing the desired qualities of SOA	**Standard Interface**
		Standard Messaging Protocol
		Stateless Services
		Separation of Concern
		Business Orientation
Best Practices	Practices which are not indispensable in pursuing the desired qualities of SOA but greatly complement and facilitate it	**Asynchronous Messaging**
		Standards Based
		Message Bus / Enterprise Service Bus
		Registry / Repository
		Meta Data Orientation

CGI

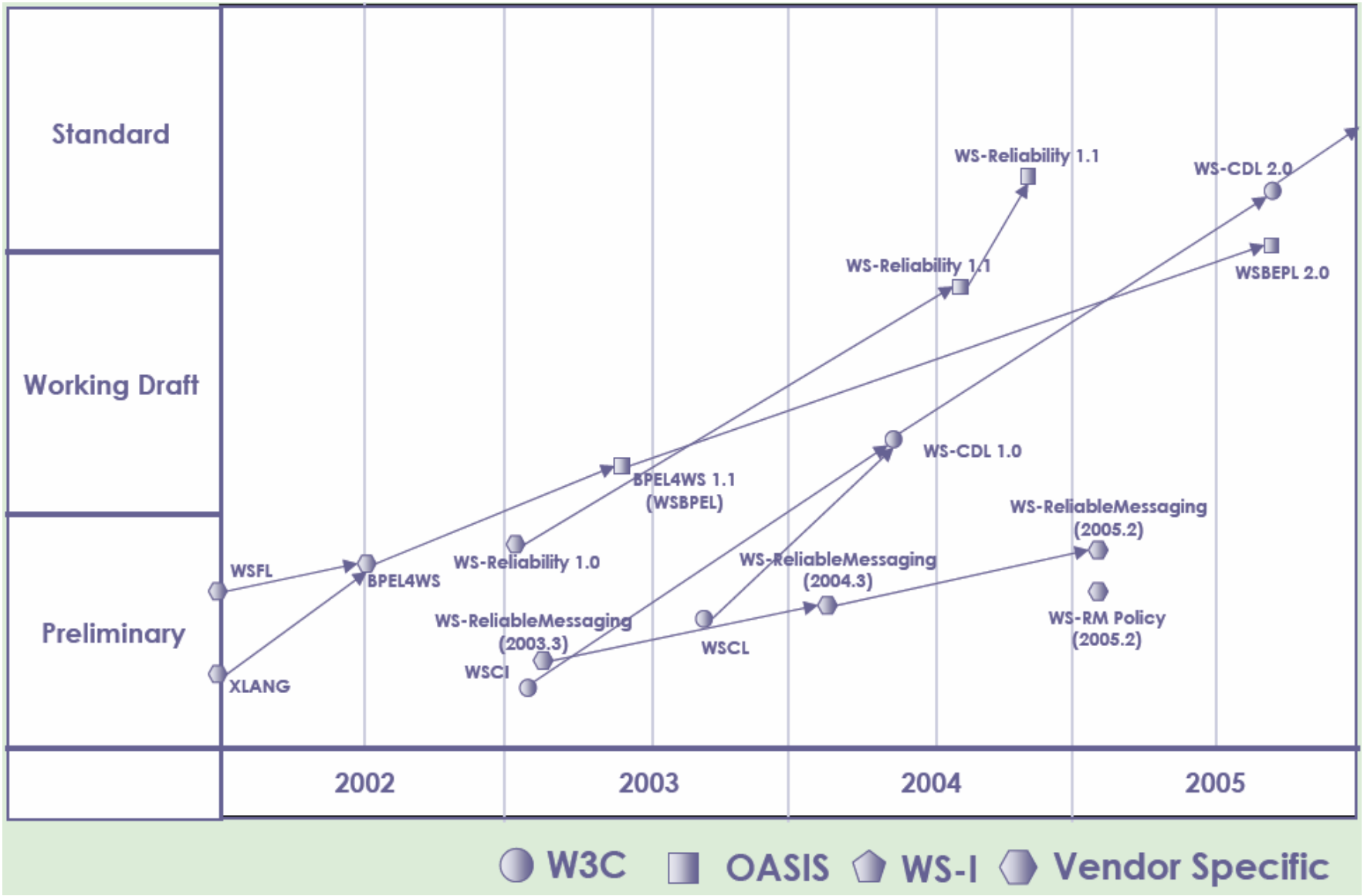
Interoperability . . .



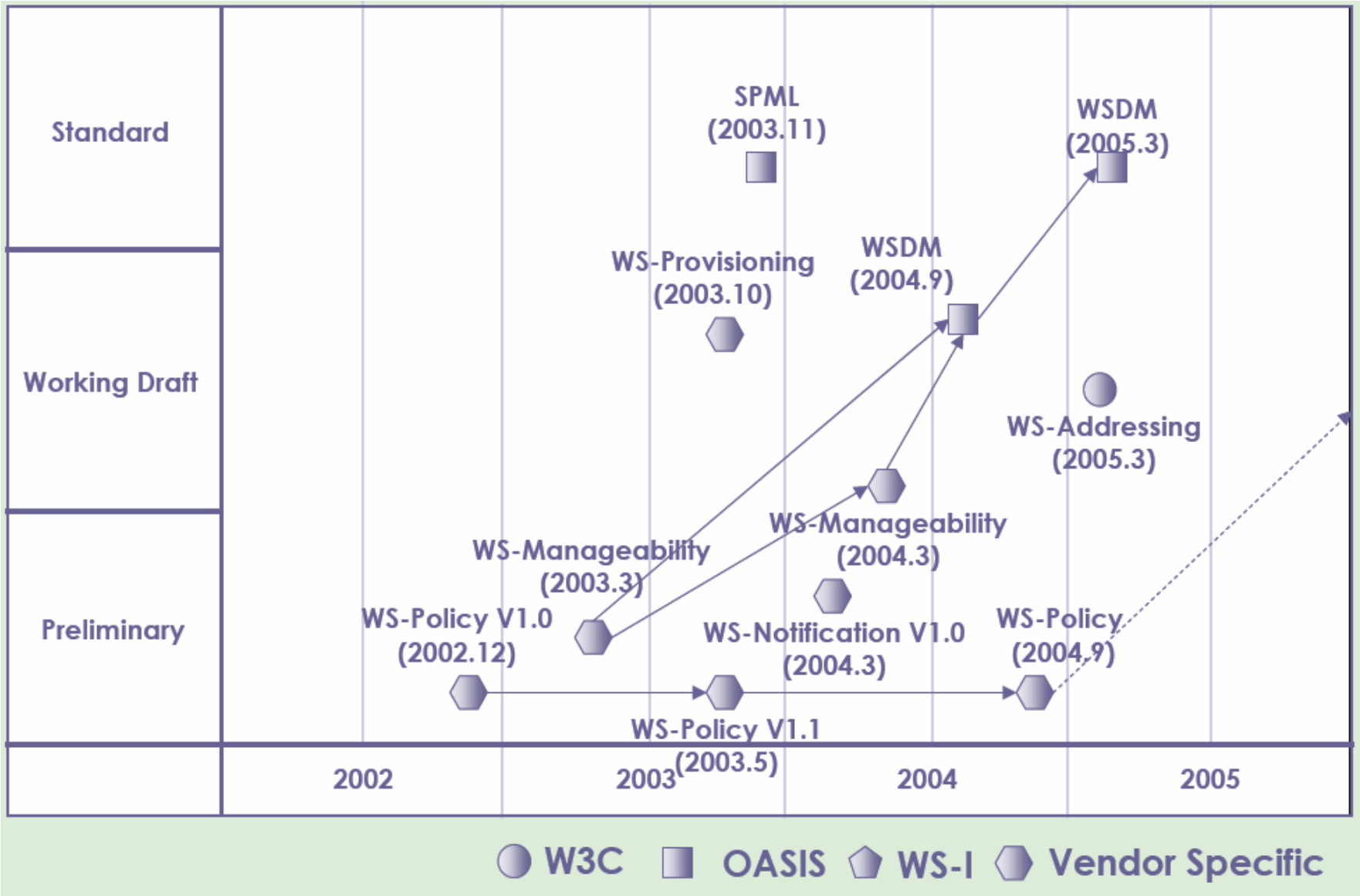
Business Processing . . .



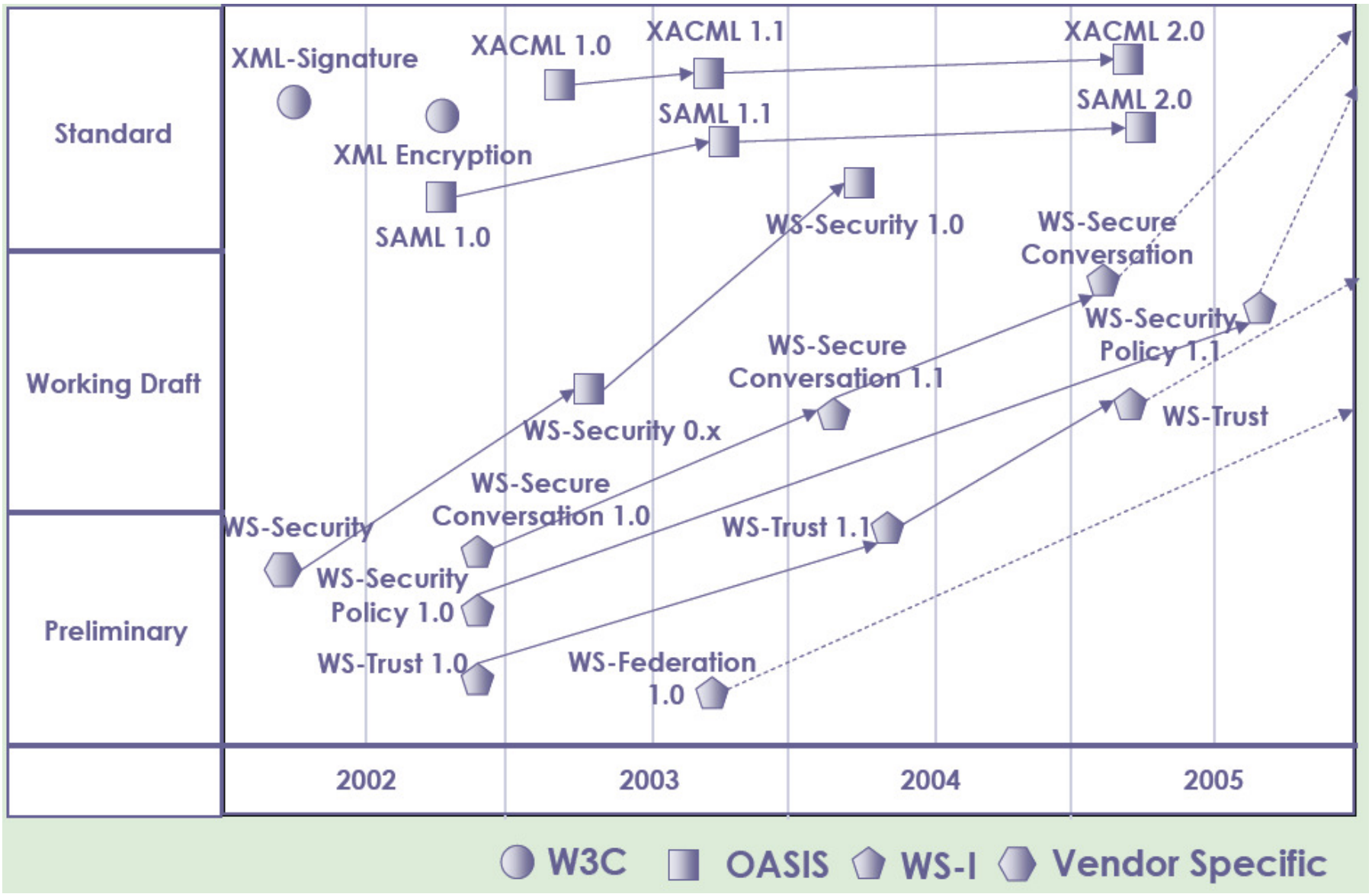
Business Processing . . . continued



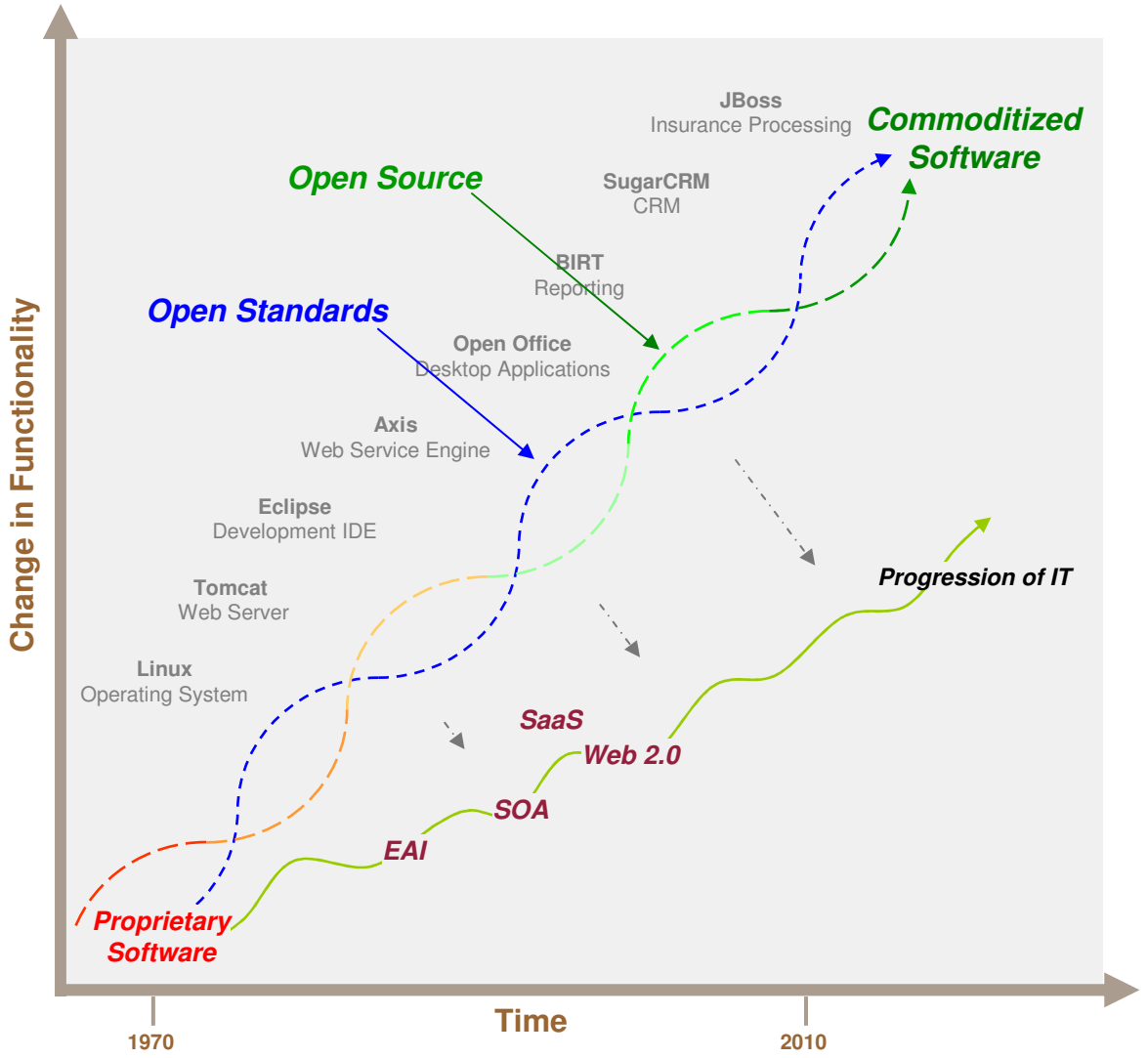
Manageability . . .



Security . . .


Standard
Working Draft
Preliminary
XML-Signature
XML Encryption
SAML 1.0
XACML 1.0
XACML 1.1
XACML 2.0
SAML 1.1
SAML 2.0
WS-Security 1.0
WS-Secure Conversation
WS-Security Policy 1.1
WS-Trust
WS-Secure Conversation 1.1
WS-Security 0.x
WS-Secure Conversation 1.0
WS-Security
WS-Trust 1.1
WS-Security Policy 1.0
WS-Trust 1.0
WS-Federation 1.0
2002
2003
2004
2005
W3C
OASIS
WS-I
Vendor Specific

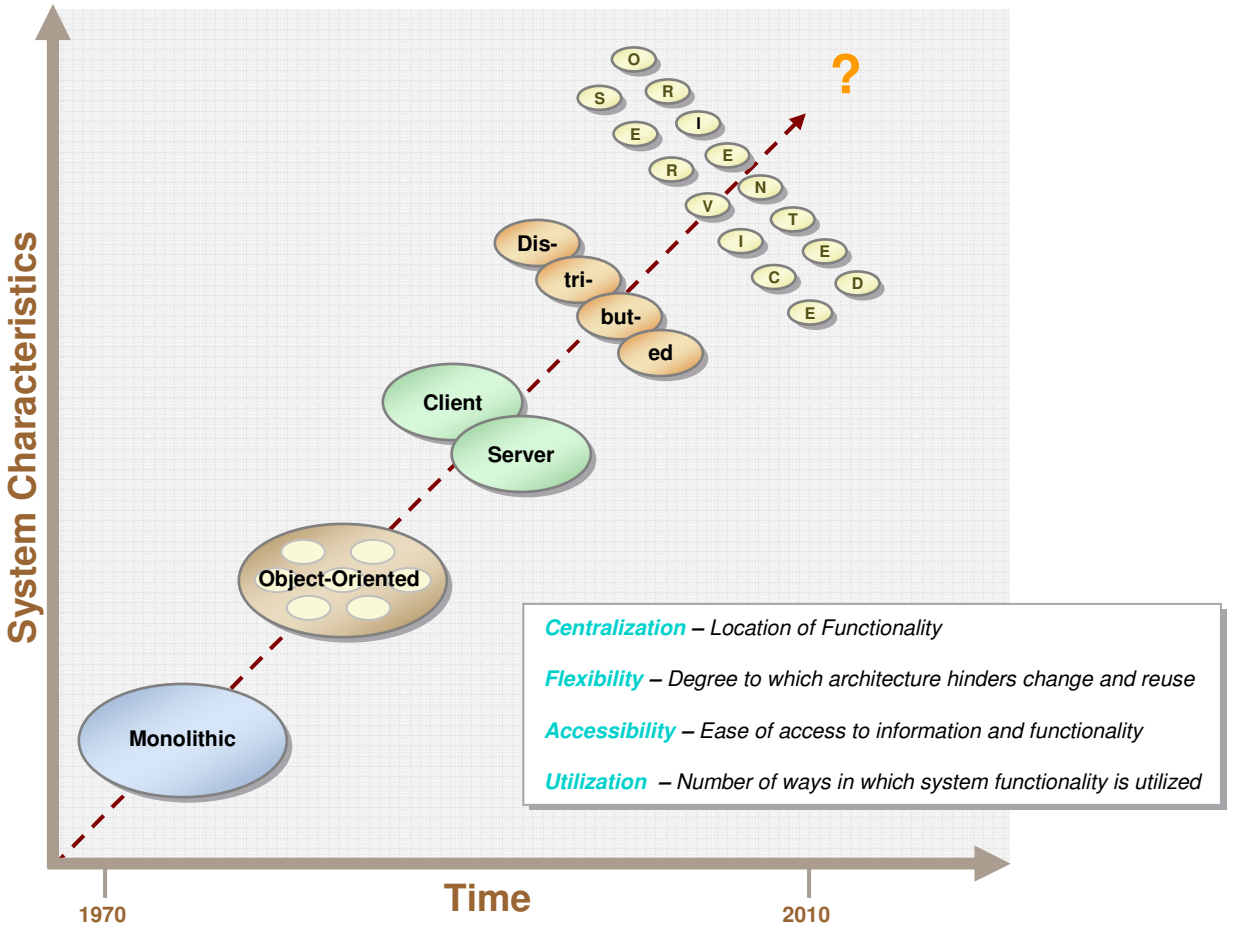
Technology Drivers . . . feeding off each other



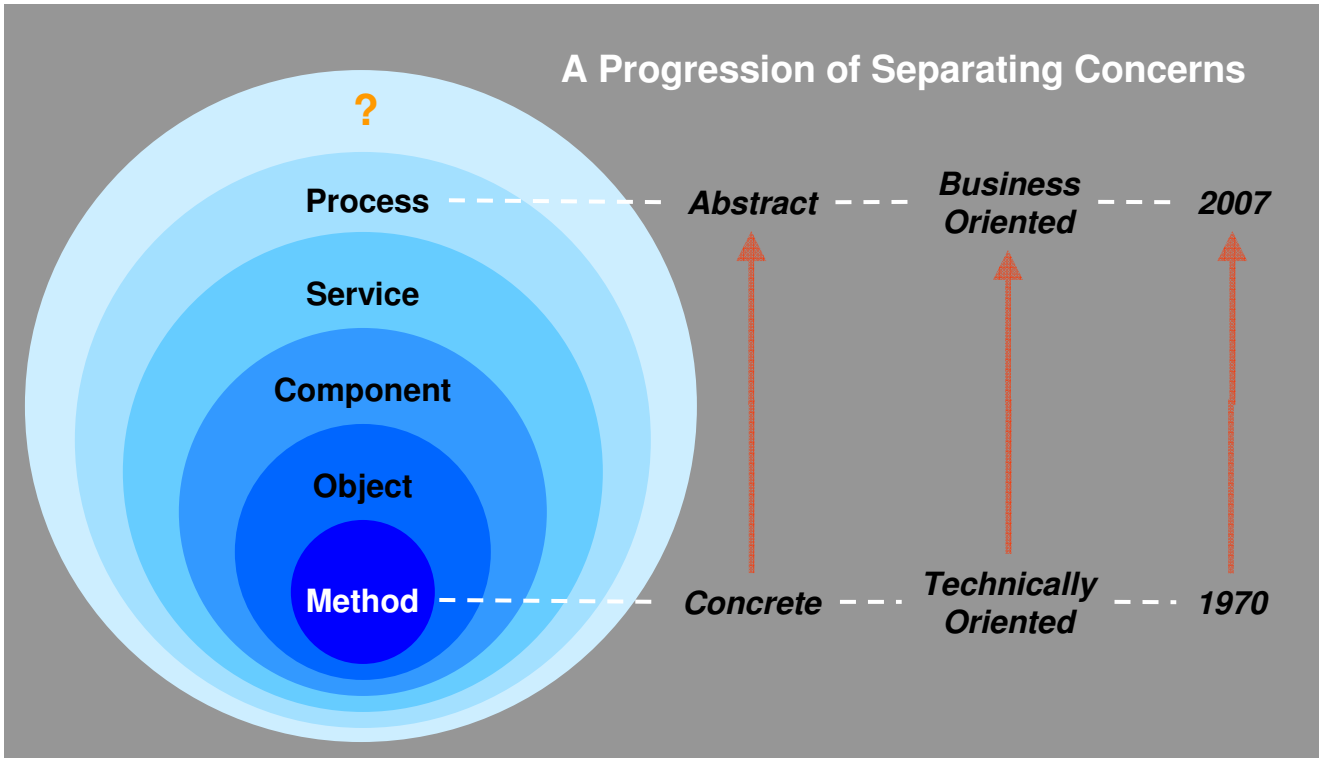
Evolution of Systems . . .

- *Decentralized*
- *Flexible*
- *High Accessibility*
- *Wide Utilization*
- *Standards Based*

- *Centralized*
- *Inflexible*
- *Low Accessibility*
- *Narrow Utilization*
- *Proprietary*



An Evolution of Encapsulation . . .



"The principles of business components and SOA have been understood since the 1990s, but the implementation is better in 2005, because of the growing use of industry standards (especially Web services), BPM and EDA. "

– Roy Schulte, Gartner Group

The Bad

Misconceptions . . .

- Misconceptions continue to Abound:
  - SOA is a **product that can be bought**
  - SOA is **just a set of WSDL interfaces** to an application
  - SOA is **just web services**
  - SOA is **EAI at a B2B Level**
  - **UDDI is essential** to SOA
  - WSDL **is an effective means of achieving loose coupling**
  - **Governance is the critical hurdle** for the masses **in** achieving effective SOA
  - **Web Services Standards are done evolving**
  - Data must reside in a **relation database**
  - . . . etc

CGI

Disagreements . . .

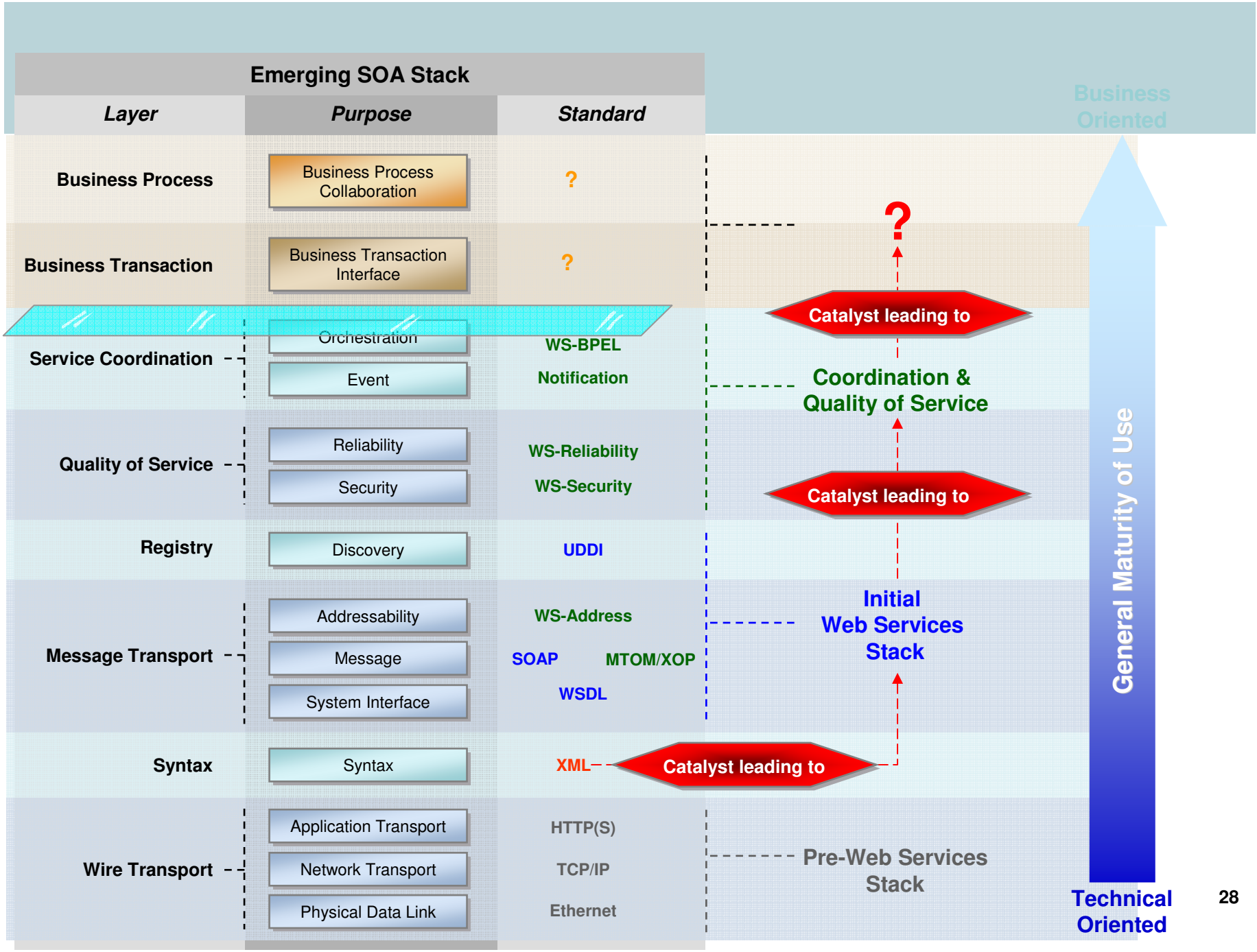
- Disagreements continue to Flourish . . .
  - While some vendors may agree on some standards, few are defining SOA or their products the same.
    - **Standards Based** (fundamental to an open architecture)
    - **Standards Friendly** (extensions to a proprietary architecture)
    - **Standards Submerged** (proprietary architecture enveloping an open architecture)
    - Call it a **Bus** but sell a **Broker**
    - Call it an **Open Standards based Registry** but sell a **proprietary repository**
    - **. . .** etc

SOA Implications . . . An Evolution not a Revolution

- **Products Are Emerging (rapidly)**
  - Enterprise Service Bus
  - BPEL Engines
  - Business Activity Monitors
  - Etc . . .
- **Standards Are Evolving (slowly and not always smoothly)**
  - Maturing
    - WS-I Profiles - Standards for how to use standards are emerging
    - WSDL 1.1 > 2.0
    - BPEL 1.1 > 2.0
  - Consolidating
    - Redundant/conflicting standards being left behind
- **Practices Are Emerging (slowly)**
  - Granularity of Services
  - Organization of Services
  - Managing Services
  - . . .

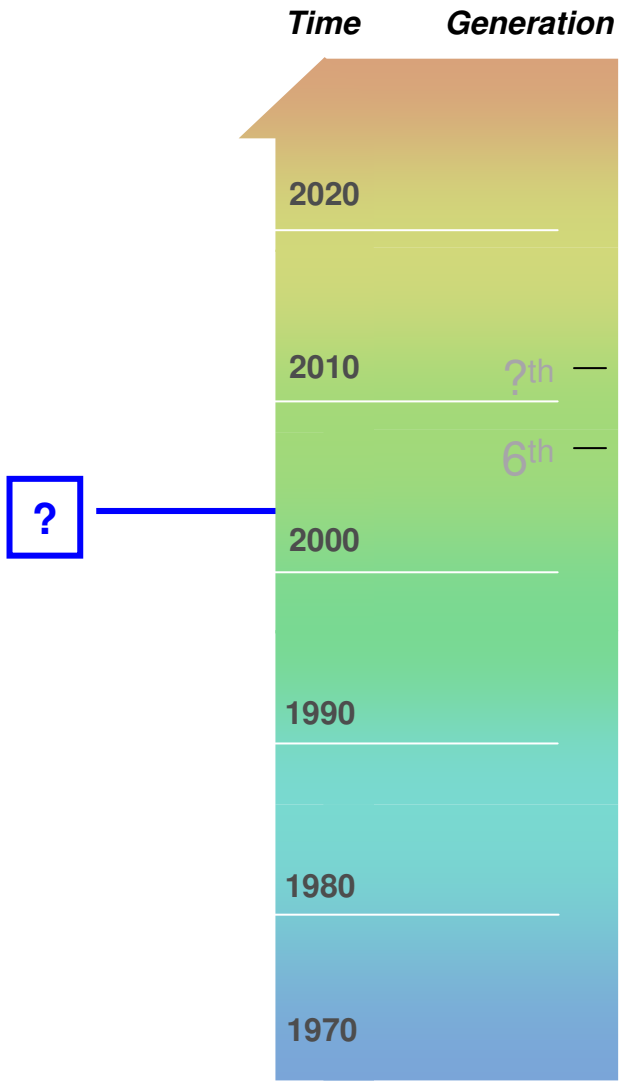
CGI

The Ugly


Emerging SOA Stack
Layer
Purpose
Standard
Business Oriented
Business Process
Business Process Collaboration
?
Business Transaction
Business Transaction Interface
?
?
Catalyst leading to
Service Coordination
Orchestration
WS-BPEL
Event
Notification
Coordination & Quality of Service
Quality of Service
Reliability
WS-Reliability
Security
WS-Security
Catalyst leading to
Registry
Discovery
UDDI
Initial Web Services Stack
Message Transport
Addressability
WS-Address
Message
SOAP
MTOM/XOP
System Interface
WSDL
Syntax
Syntax
XML
Catalyst leading to
Wire Transport
Application Transport
HTTP(S)
Network Transport
TCP/IP
Physical Data Link
Ethernet
Pre-Web Services Stack
General Maturity of Use
Technical Oriented

Generations . . .

- An Evolution of:
  - **Standards**
  - **Products**
  - **Practices**



Generations . . . the Continuing Evolution of Integration

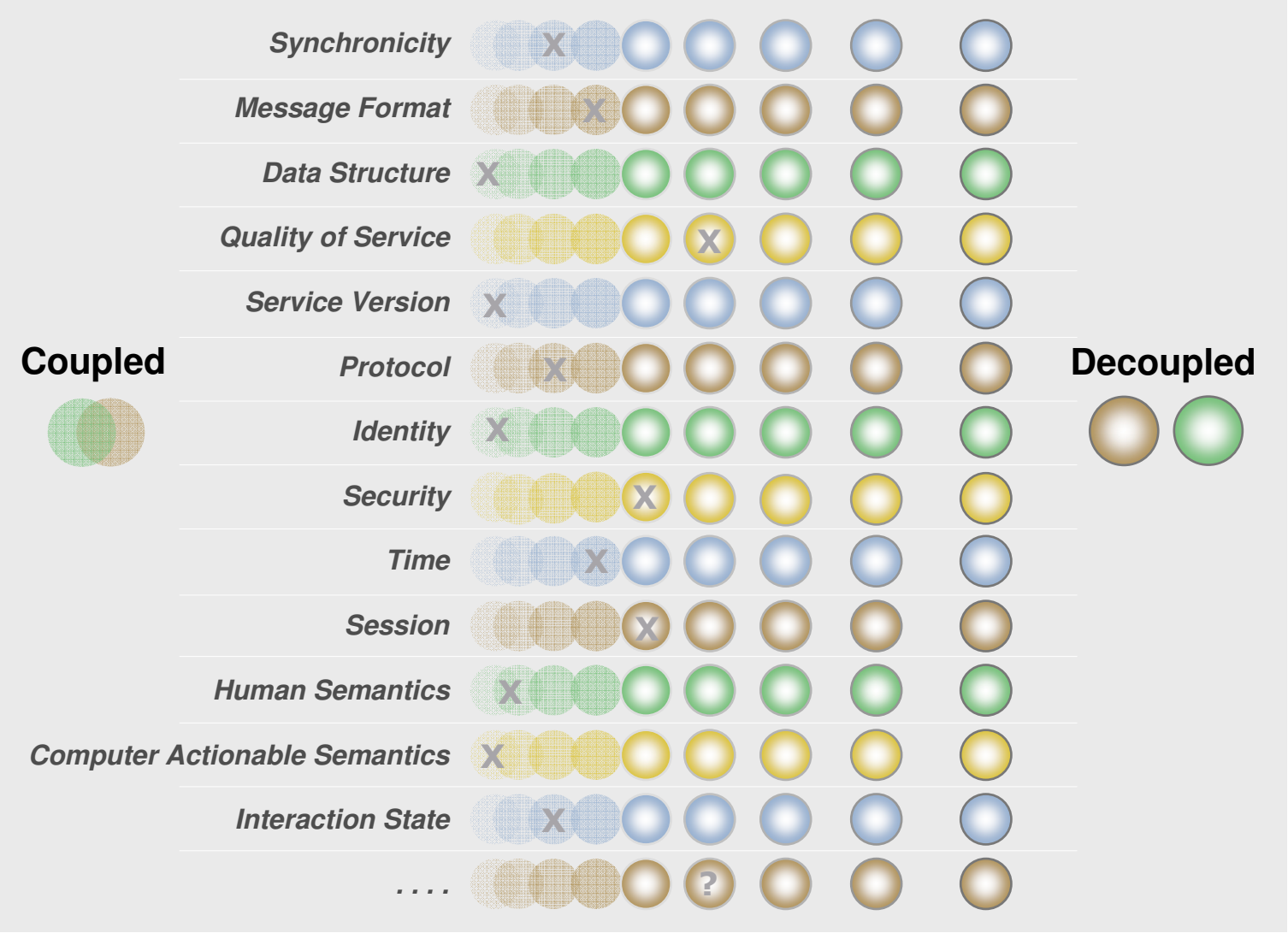
G#	Popular Name	Characteristics	Short Comings
0	Stove Pipe	• No Data Sharing	• Redundant data, No Visibility
1	Home Grown	• EDI & Sockets & DB	• Expensive to build and maintain, Brittle
2	CORBA	• 1st Generation Standards-based	• Complex, Expensive to integrate, Brittle
3	EAI	• Proprietary	• Vendor Lock-in, High coupling, Expensive to scale in breadth
4	**Web Services (Visionary)**	• **Single Universal Registry (UDDI)**	• Unrealistic
5	**Web Services (RPC)**	• **Remote Procedure Calls (without UDDI)**	• High coupling
6.1	**Web Services (Doc Literal)**	• **Message Oriented**	• Expensive to scale in breadth • Limited applicability
6.2	**Web Services (WS-I)**	• **Standard for using Standards**	• Limited scope in early versions
6.3	**Web Services (WS-Address)**	• **Self-directed Message**	• Expensive to scale in breadth
6.4	**Web Services (WS-BPEL)**	• **Standards-based Service Orchestration**	• Expensive to scale in breadth
6.5	**Web Services (WS-QoS)**	• **Quality of Service (e.g. Reliability)**	• Expensive to scale in breadth
7	?	?	?
8	. . .	. . .	. . .
9	. . .	. . .	. . .

We are here

CGI

* Abbreviated for ease of illustration

Degrees and Dimensions of Coupling . . .



Richness and Breadth . . .

Scaling for richness and breadth of functionality

How to go from effectively deploying:

15 services
to
15,000 services . . .

Service Sprawl . . ?

Governance

- Could it be that the cry for "**SOA Governance**" is largely a **euphemism** for trying to **contend with the accidental complexity arising from the immaturity** of the current web services standards and tools . . ?

Analogous to the discipline required to build an OO application in a procedural language

CGI

Barrier to Entry

For every **$1** spent on software license. . .

$5 is spent on integration

- Gartner Group

Why do we handicap our users?

1. **View of Information**
   - Only one way to look at information . . ?
2. **Utilization of Information**
   - Predetermined at design time . . ?
3. **Quality of Information**
   - SOA has brought attention to the quality of transport . . . what about the quality of information . . ?
4. **Etc . . .**

Why do we handicap our system architectures?

"**Growing complexity in an enterprise's systems can fossilize operations**"

- Jeanne Ross et al, **Enterprise Architecture as Strategy: Creating a Foundation for Business Execution**, Harvard Business School Press

"Complexity is the most serious inherent difficulty, but not all complexity is inevitable. **Much of the conceptual complexity in our software is arbitrary** . . . In my experience, most complexities which are encountered in systems work are symptoms of organizational malfunctions. **Trying to model this reality with equally complex programs is actually to conserve the mess instead of solving the problems.**"

- Fred Brooks, **The Mythical Man-Month**

CGI

Why do we handicap our computers . . ?

1. **Identification** of Data
   - Implicit only . . ?
2. **Meaning** of data
   - Significance of data consistently interpreted, human aware only, and correct utilized . . ?
3. **Meta Data**
   - In the database only . . ?
4. **Purpose of Structure** . . ?
   - For presentation only . . ?
5. **Standards**
   - Benefit only if rigid and prescriptive . . ?
6. **Usability**
   - Only relevant to the UI . . ?

What are we assuming about SOA. . ?

- The standards are **done evolving**?
- The vendor products **are sufficient**?
- The necessary and fundamental practices have been **identified**?

We can build a resilient architecture with the tools and open standards we have today by just using them using them as advertised?

CGI

Breaking Through ?

What if . . ?

- **What if** we could lower the cost of access to sharing rich interactive data?

For every **$1** spent on software . . .

Spend $2.50 on integration . . ?

Spend $1.50 on integration . . ?

Spend 75¢ on integration . . ?

Spend 25¢ on integration . . ?

…

CGI

What if . . ?

- **What if** it were possible to create something that analogous to what the "**Browser**" did for the internet but in the problem space of easily and reliably sharing rich information between:
  - **Diverse organizations**
  - **Diverse systems**
  - **Diverse users**

. . . all in a rapidly changing world?

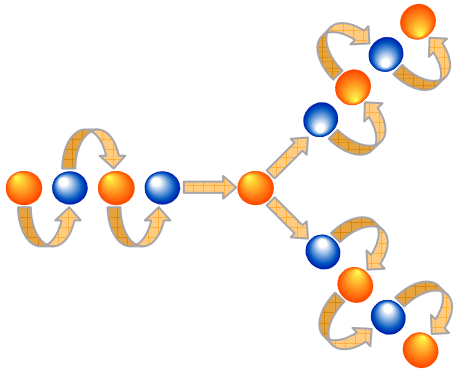
How would you do it . . ?
What would it look like . . ?

CGI

An Information Supply Chain . . .

An **Information Supply Chain** – Transcending Application Integration to a Value Chain of Information

- **Analogous** to Materials Supply Chain from manufacturing and retail world
- **Extends** on the Porter Value Chain Model
- **Transcends** the current market hype on SOA to a next generation set of issues > Semantics

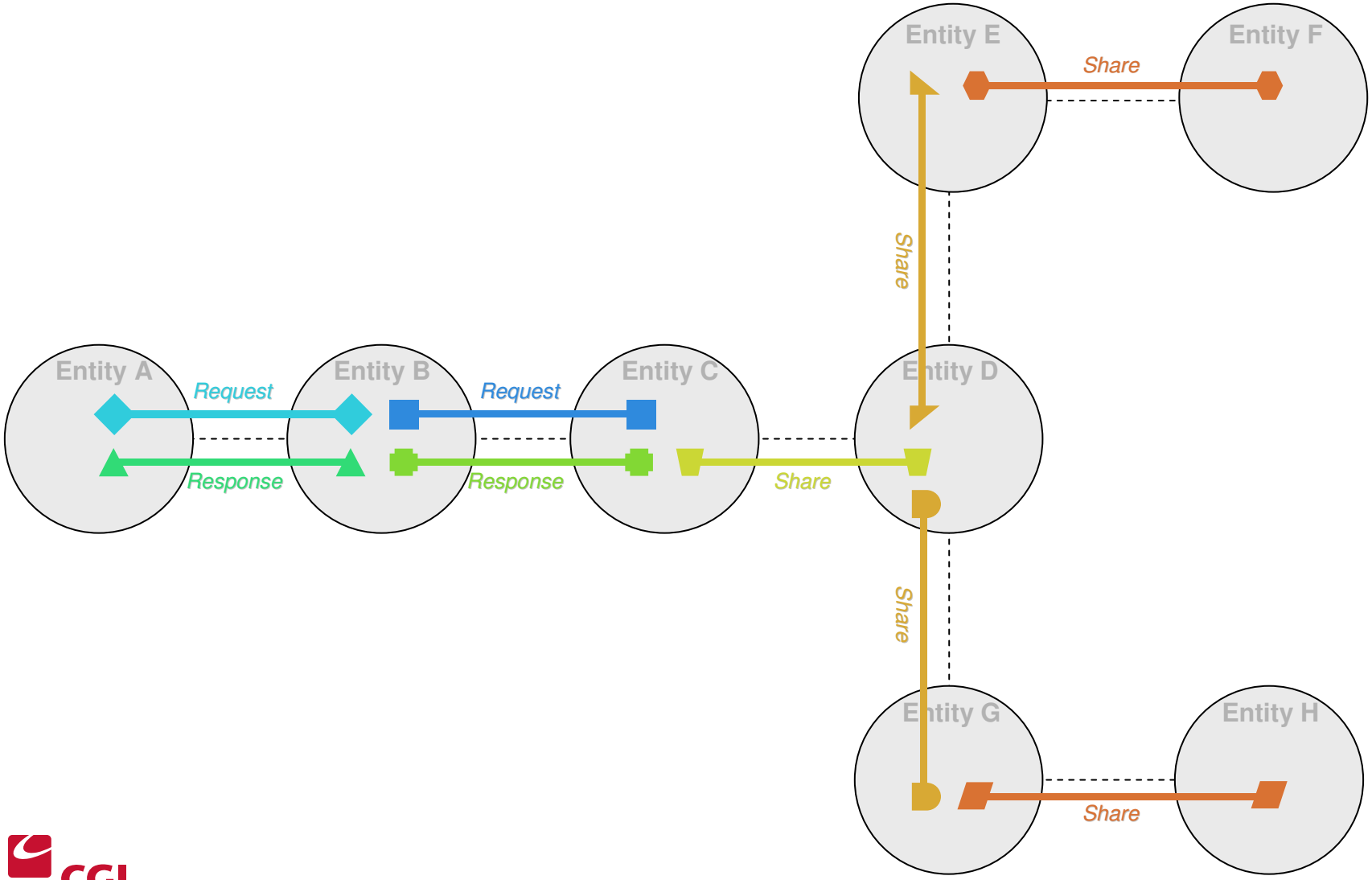


CGI

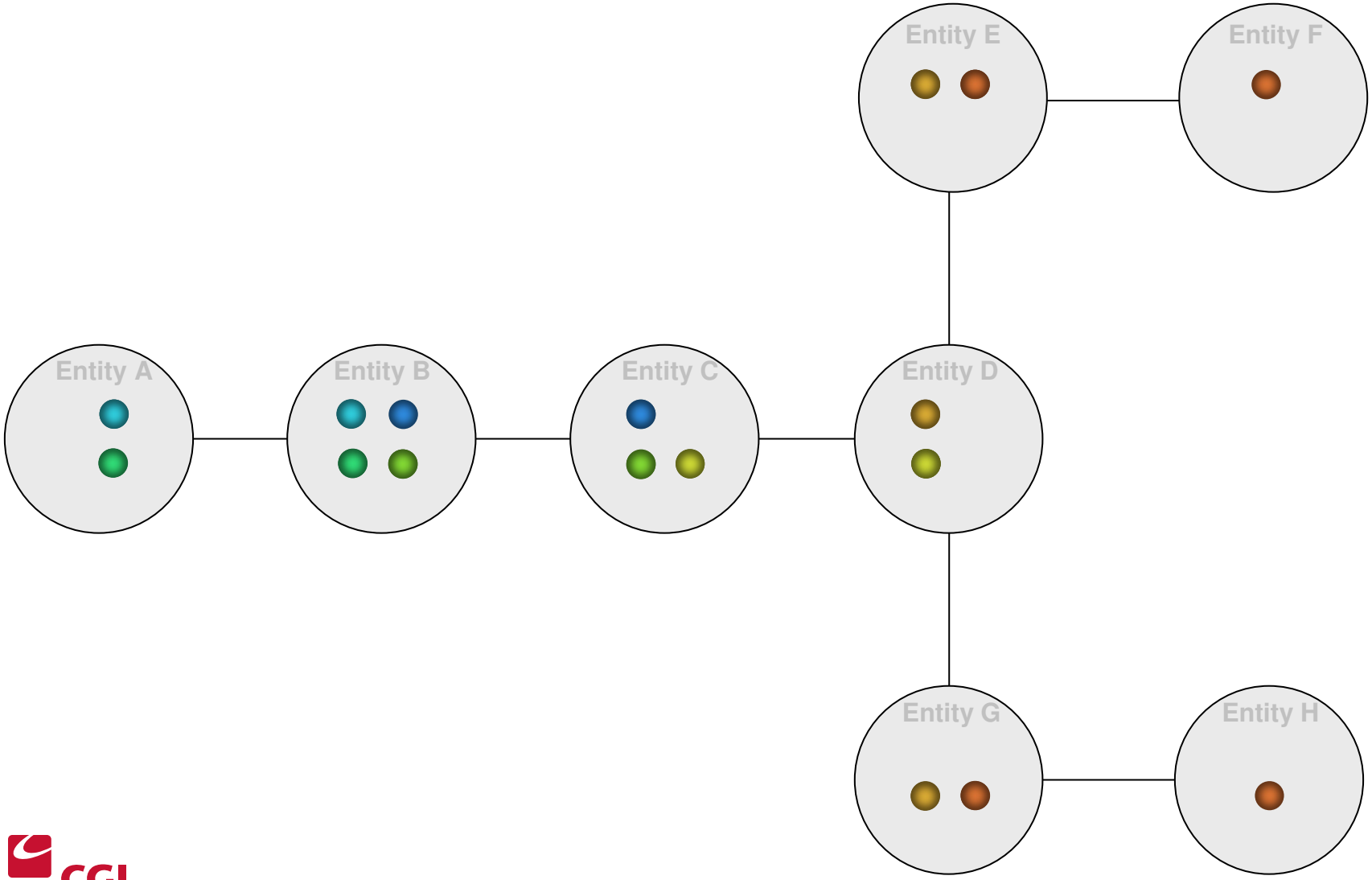
Reuse and diversity . . .

An Information Supply Chain emphasizes the notion that information can be ‘reused’ by numerous interested parties for more than one purpose and in diverse ways.

Point to Point . . .



An Information Supply Chain . . .

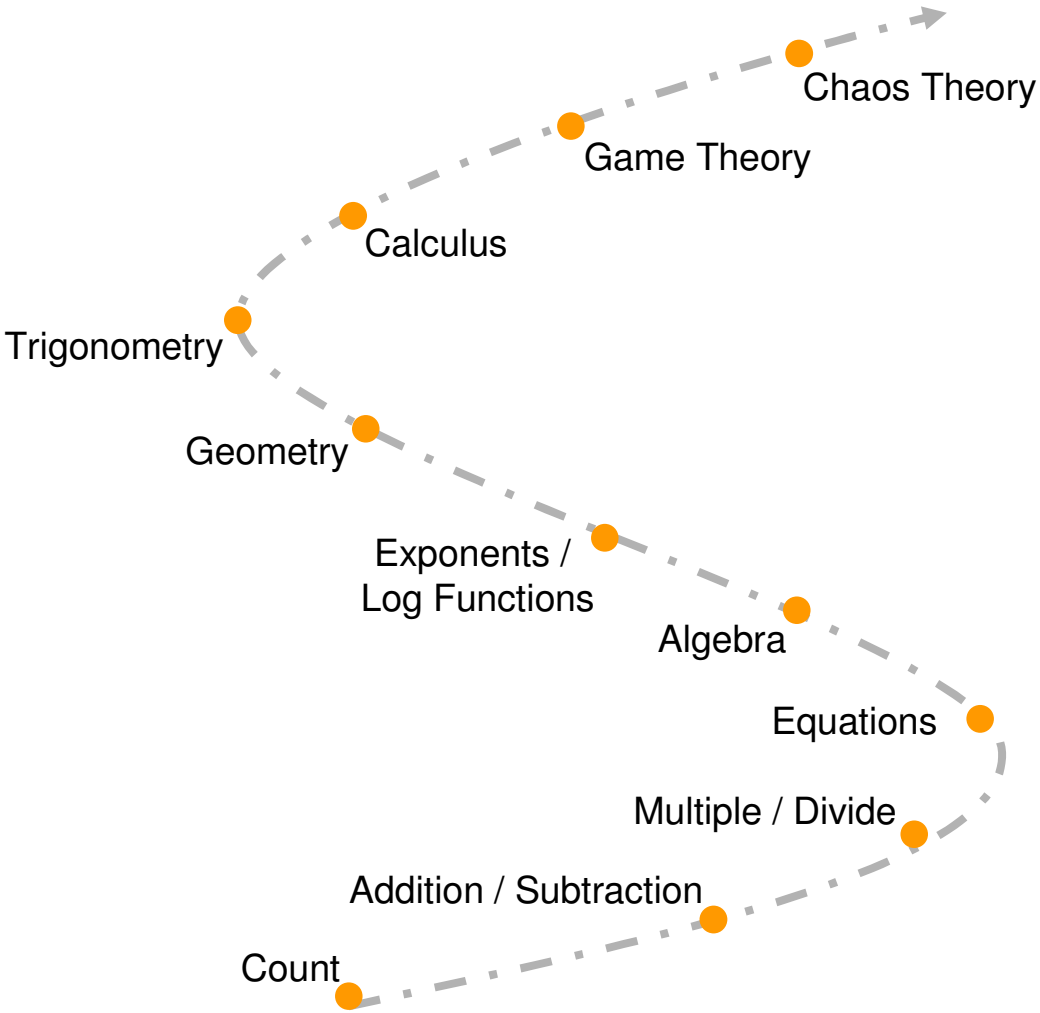


Information Supply Chain

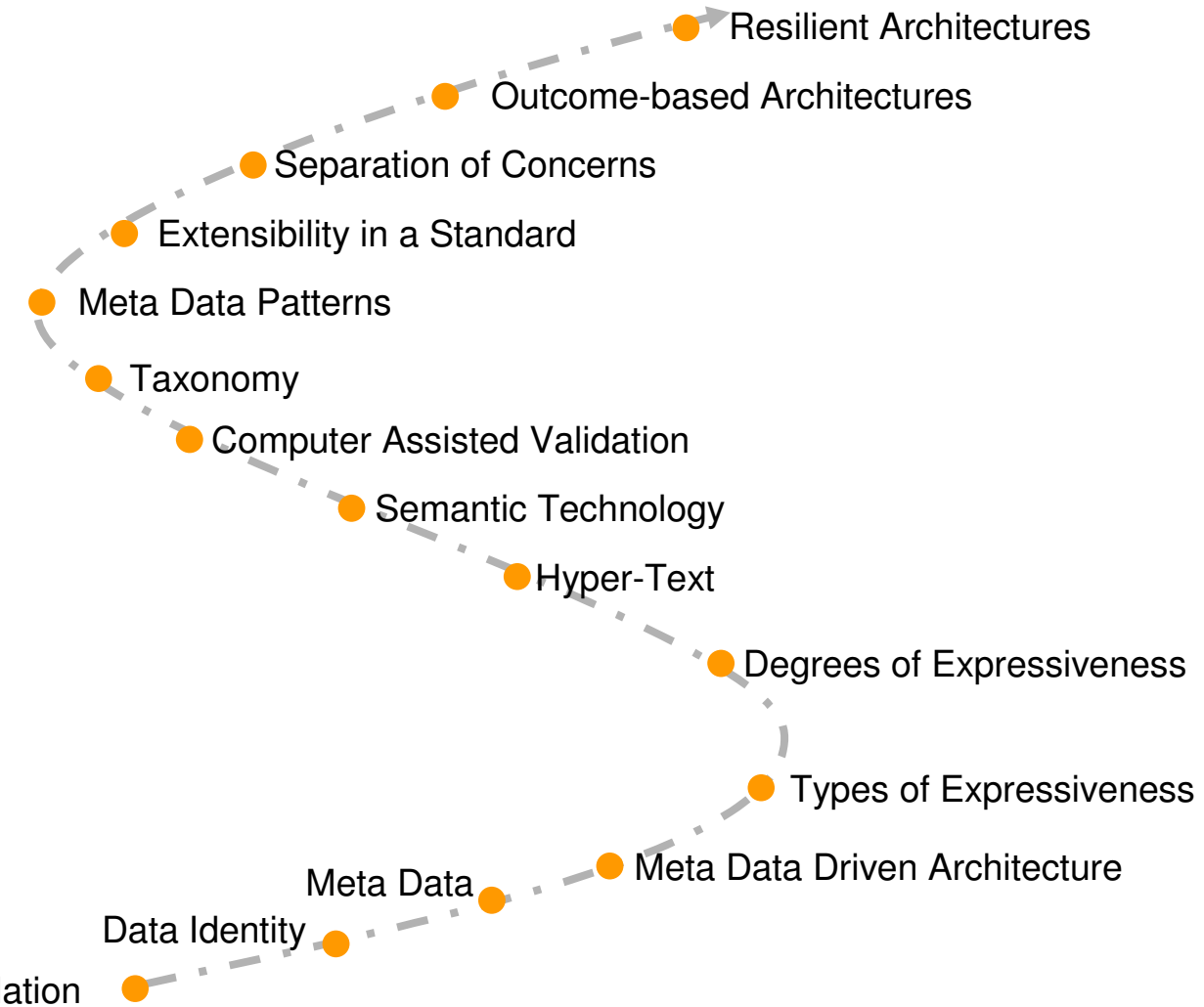
In a modern era, financial reporting is a fluid dynamic business function seeking to **leverage technology, distribute costs, and employ common semantics** - as a communal need of many organizations with **a common interest**.

A Progression of Concepts

A Progression of Concepts . . .



A Progression of Concepts . . . IT Architecture



Encapsulation

1

Encapsulation

The process of enclosing programming elements within a larger, more abstract entity and behind a **narrowly defined means of access**

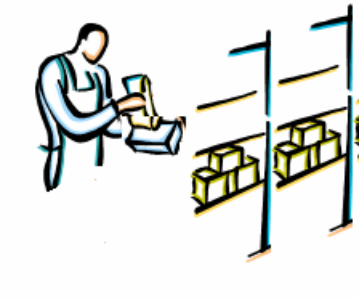
Cohesion
Indicates the degree to which things relate and thus belong together

Coupling
Indicates the degree to which unrelated things are independent of one another and thus belong apart

CGI

Significance of Identity

- Information is analogous to Products in consumer world . . .
- Grocery Stores in the 1960's
  - The heavy cost of manual identification through the product lifecycle:
    - Identifying what something is . . .
    - And then taking the correct action with it . . .


Savings in the Retail World . . .

- Universal Product Code:
  - Saved an average of $44 billion per year over the 25 years.[1] (approximately $1.4 trillion by 2007)
  - Scanned over 5 billion times per day

[1] PWC study of 1974 to 1999

Common Misconception

- Because information '**appears**' structured when presented (HTML, Excel, etc) the computer can aid in the utilization of that information

Most often the computer has no notion of the significance or the nature of what it presents

Examples:
- HTML
- Excel
- Word

Examples:
- UPC
- RFID
- ISBN
- SSN

CGI

Loosing Identity . . .

Once pulled outside the application by a query or by a report writer, data is typically **stripped of all supporting information and becomes just a number**. Analysis of the how and why for a particular number on a subsequent financial report becomes an exercise of going back to the original software application and hoping that enough **identifying** information still exists inside the application. Accounting applications tend to keep supporting data in ways that are unique to each application, complicating both the business reporting process and subsequent internal control and audit routines.

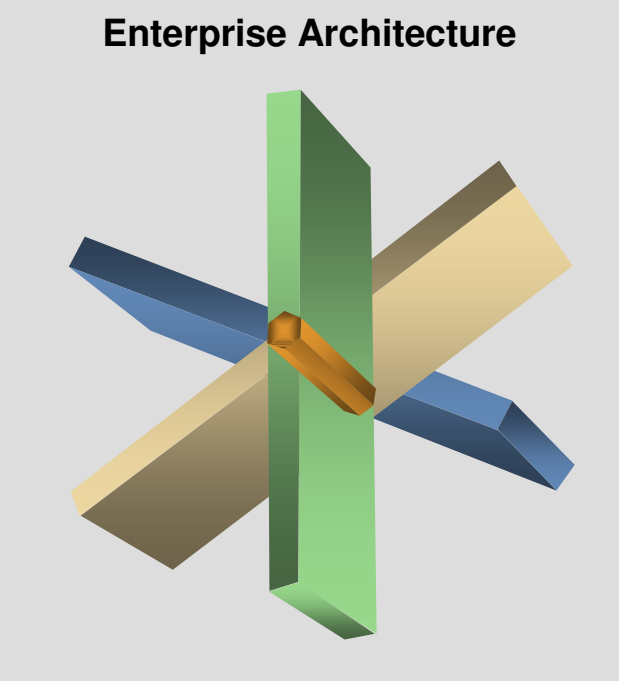
- Neal J. Hannon, Strategic Finance Magazine

Meta Data

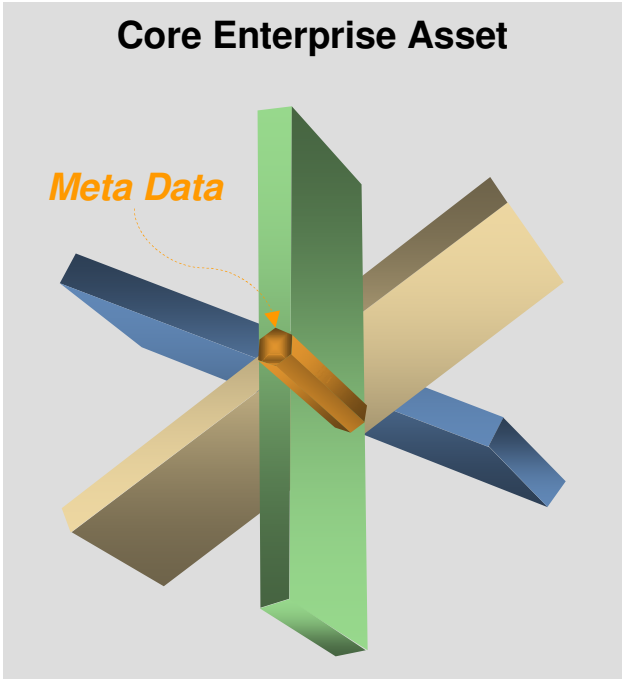
- **Information about Information**
  - For example a Dictionary has more than a single definition per word. A dictionary captures numerous types of information about that word, such as:
    - Pronunciation
    - Function (e.g. verb, noun, adjective, etc)
    - Etymology
    - History
    - Synonym
    - Antonym

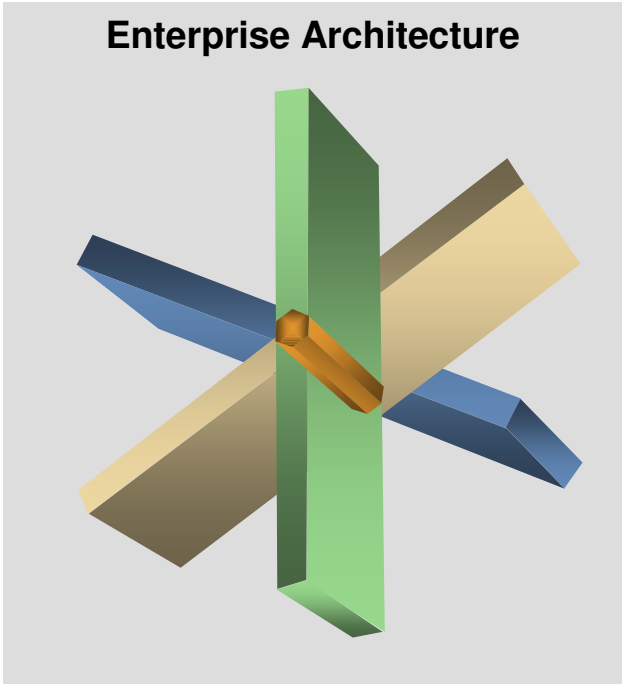
Enterprise Architecture . . .

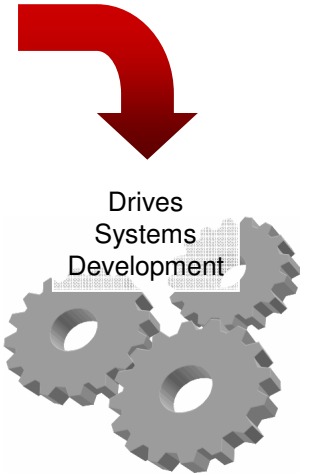
The Intersection of 3 Dimensions



Information Architecture





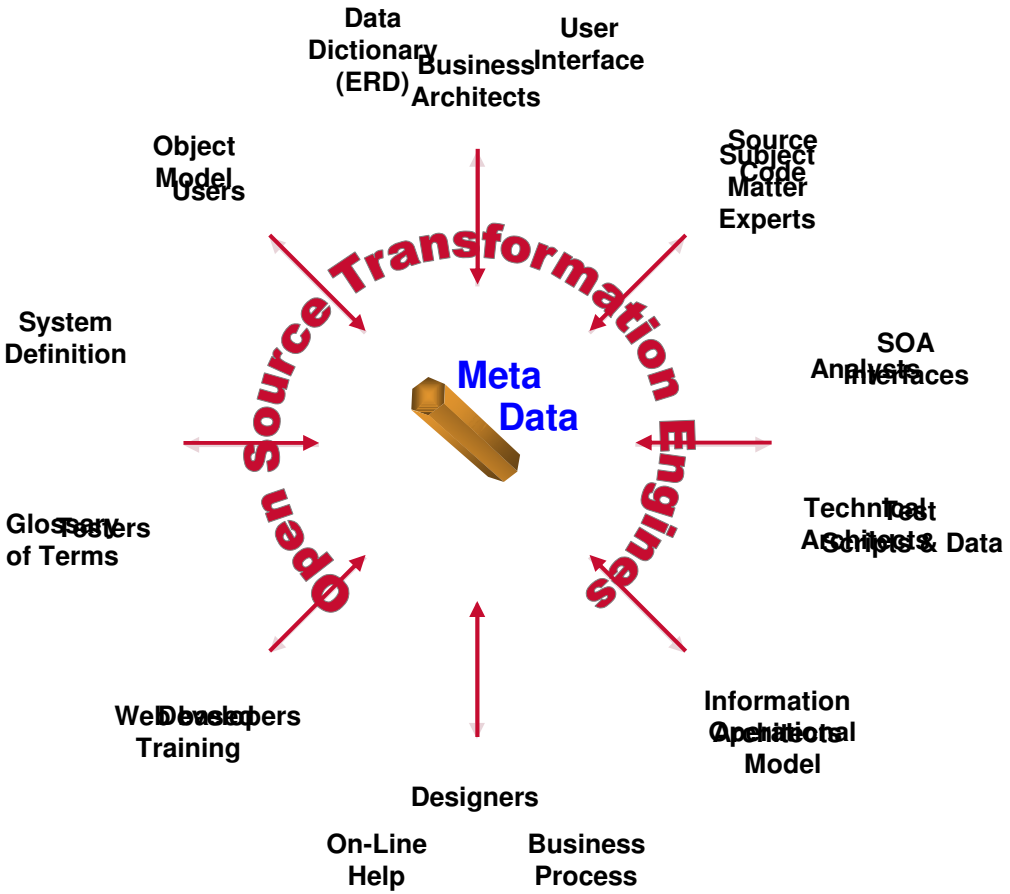


Realizing the Full Potential of a key Enterprise Asset . . .
Meta Data: Information about your information

Meta Data Driven Architecture

At the Center of Everything

Meta Data Driven Architecture



Types of Expressiveness

Implicit

to involve or indicate by inference, association, or necessary consequence rather than by direct statement

NOT ACTIONABLE without human interpretation or custom coding

Explicit

fully revealed or expressed without vagueness, implication, or ambiguity: *leaving no question as to meaning or intent*

ACTIONABLE without human interpretation or custom coding

Syntax:
- XML
- HTML
- Excel
- Word

versus

Semantics:
- **?**

CGI

Syntax verse Semantics

Definition . . . Wikipedia

Semantic technology

From Wikipedia, the free encyclopedia

In software, **semantic technology** encode meanings separately from data and content files, and separately from application code.

This enables machines as well as people to understand, share and reason with them at execution time. With semantic technologies, adding, changing and implementing new relationships or interconnecting programs in a different way can be just as simple as changing the external model that these programs share.

With traditional information technology, on the other hand, meanings and relationships must be predefined and "hard wired" into data formats and the application program code at design time. This means that when something changes, previously unexchanged information needs to be exchanged, or two programs need to interoperate in a new way, the humans must get involved.

Off-line, the parties must define and communicate between them the knowledge needed to make the change, and then recode the data structures and program logic to accommodate it, and then apply these changes to the database and the application. Then, and only then, can they implement the changes.

Semantic technologies are "meaning-centered." They include tools for:

- autorecognition of topics and concepts,
- information and meaning extraction, and
- categorization.

CGI

Semantics . . .

"Most of the Web's content today is designed for humans to read, not for computer programs to manipulate meaningfully. Computers can adeptly parse Web pages for layout and routine processing—here a header, there a link to another page—but in general, ***computers have no reliable way to process the semantics [meaning of what is on a page].***"

- Tim Berners-Lee

CGI

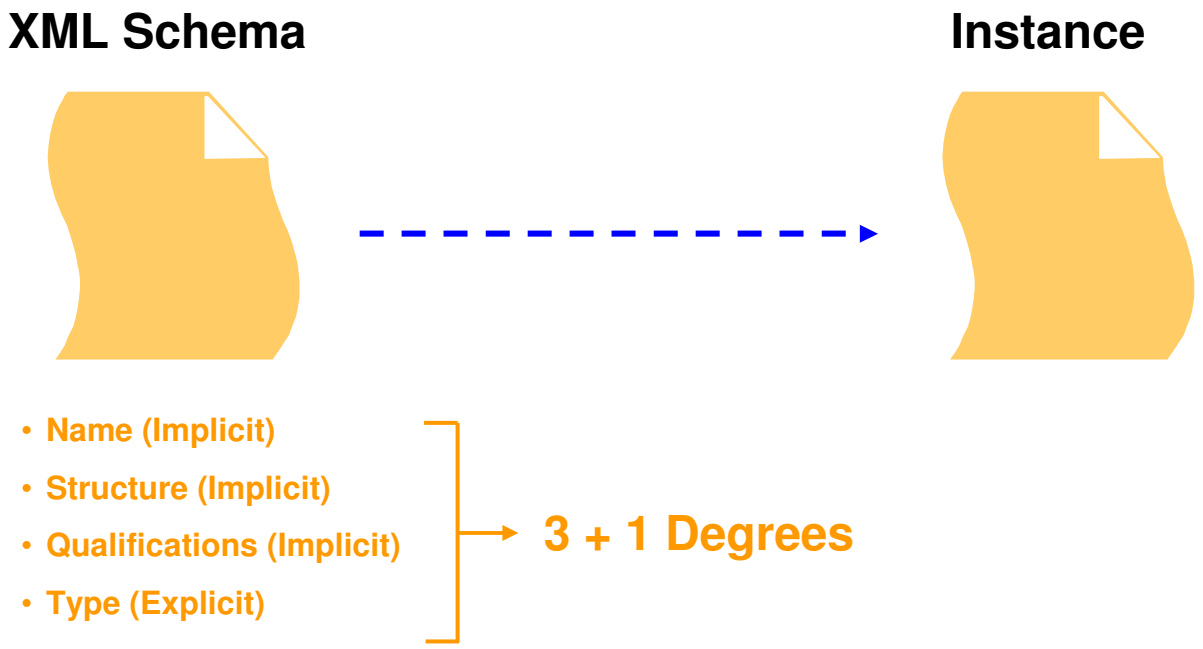
Semantic Technology . . .

- Taxonomies do contain a small amount of material that assists a syntactic validation of an instance document - the data type and the like of an element.
- However, the far greater amount of content in a taxonomy is non-syntactic. It is meaning or semantics either in relation to:
  - **The outside world**
    - Labels
    - Presentation
    - References
    - Rendering
  - **Self-consistency**
    - Definition
    - Calculation
    - Formula

Semantic Technology

- Types of Semantic Technology
  - Inference
  - Discovery
  - Data Quality/Validation
  - Data Identity
  - Assisted Manipulation
  - Automated Processing

XML Schema View of Meta Data

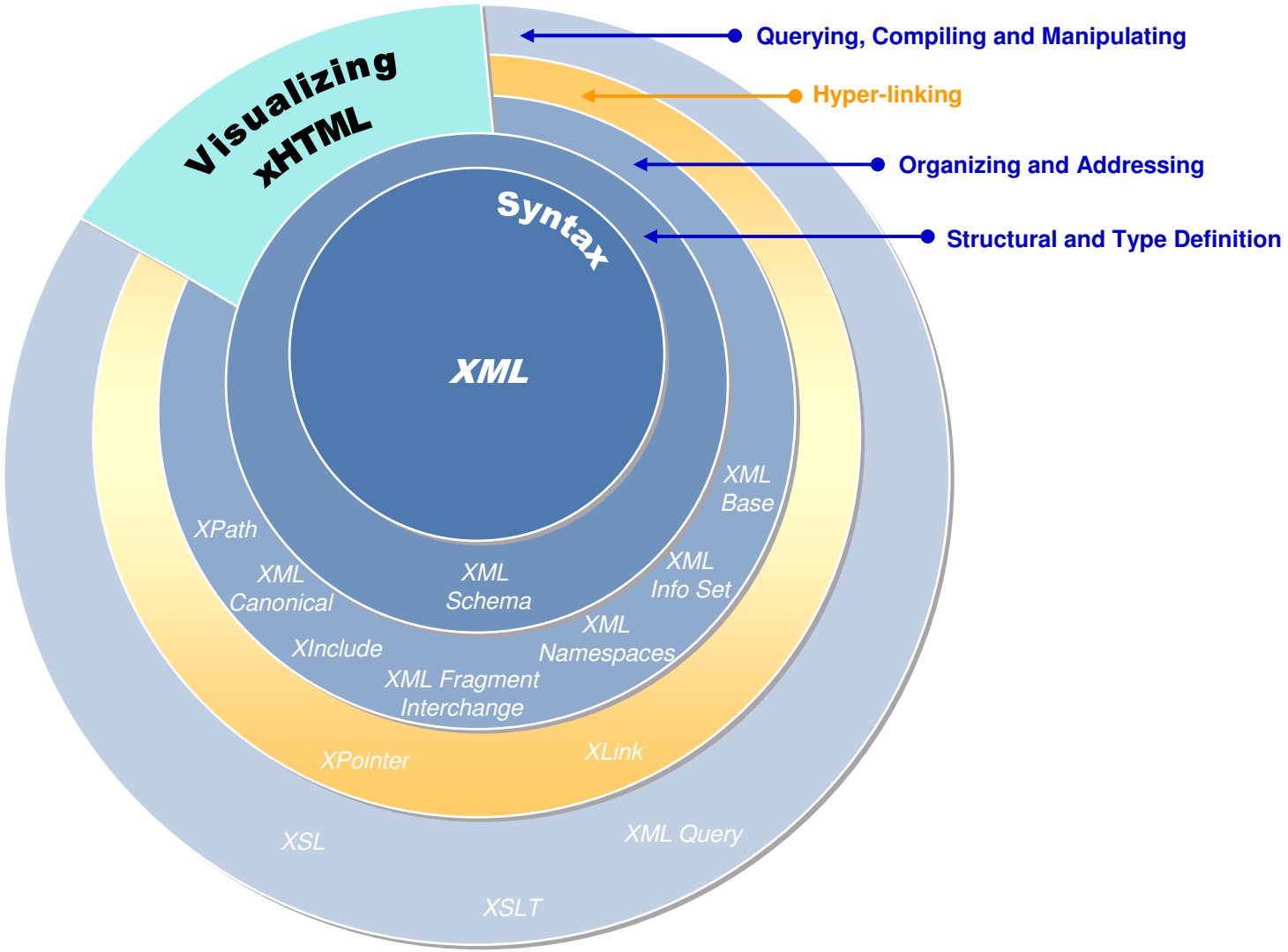


Degrees of Expressiveness

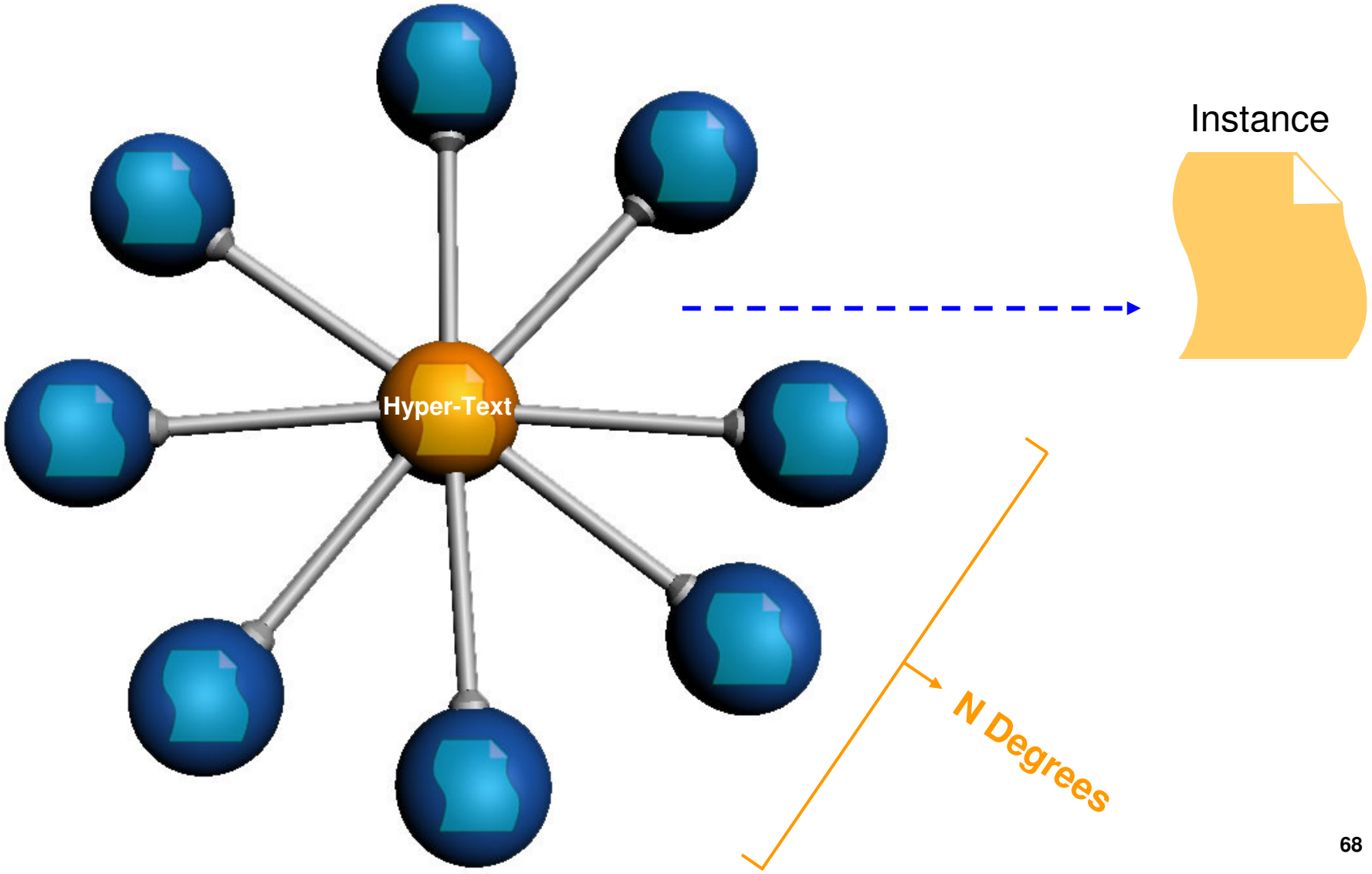
- **4 Degrees - as commonly practiced with base XML Specification**
  - Name (Implicit)
  - Structure (Implicit)
  - Qualifications (Implicit)
  - Type (Explicit)
- **N Degrees - as the original intention of the full Family of XML Specifications**
  - Via Hyper-Text

Quality and level of reuse of Information

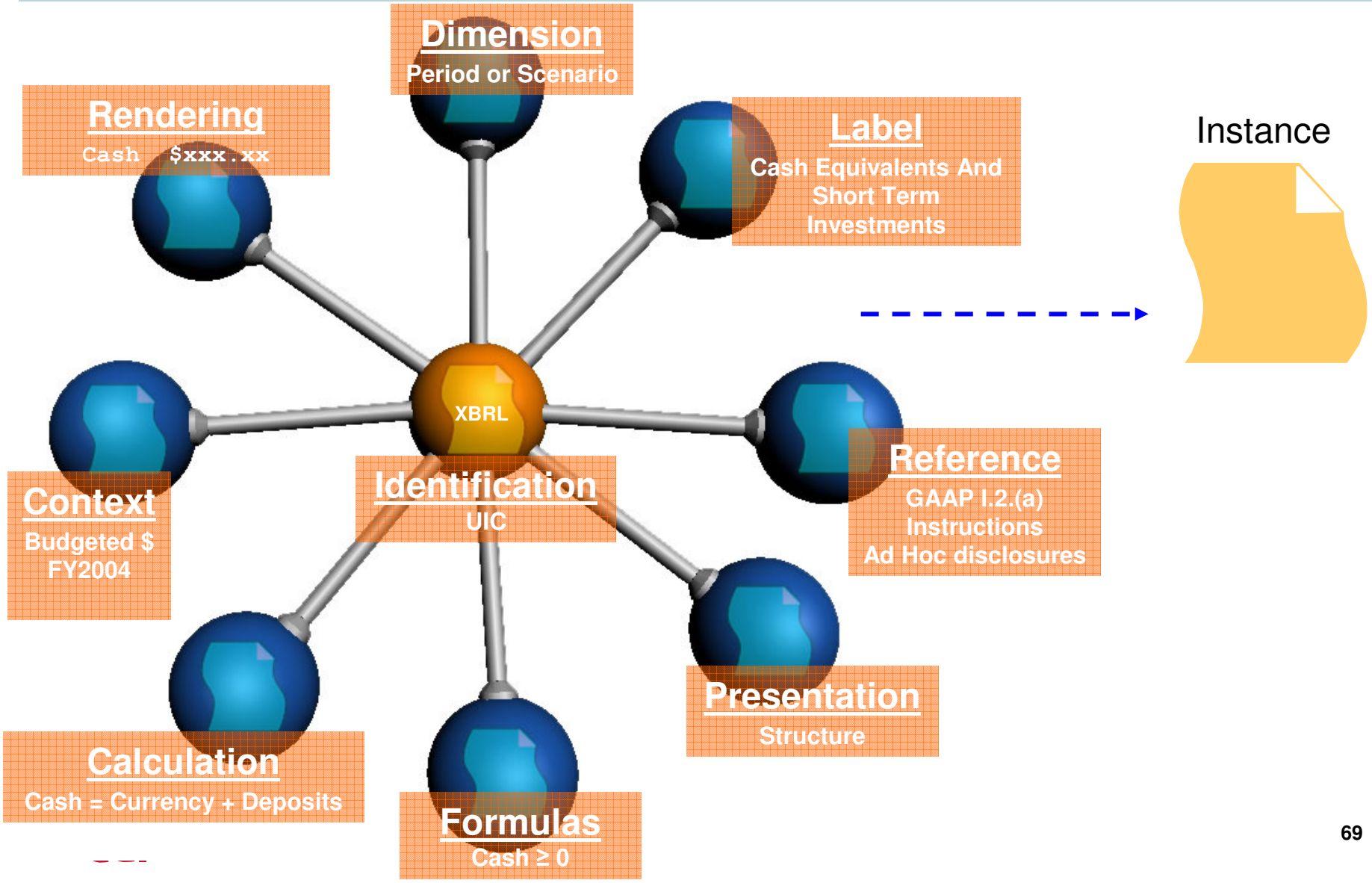
XML – A Family of Specifications



Hyper-Text View of Meta Data



Hyper-Text View of Meta Data

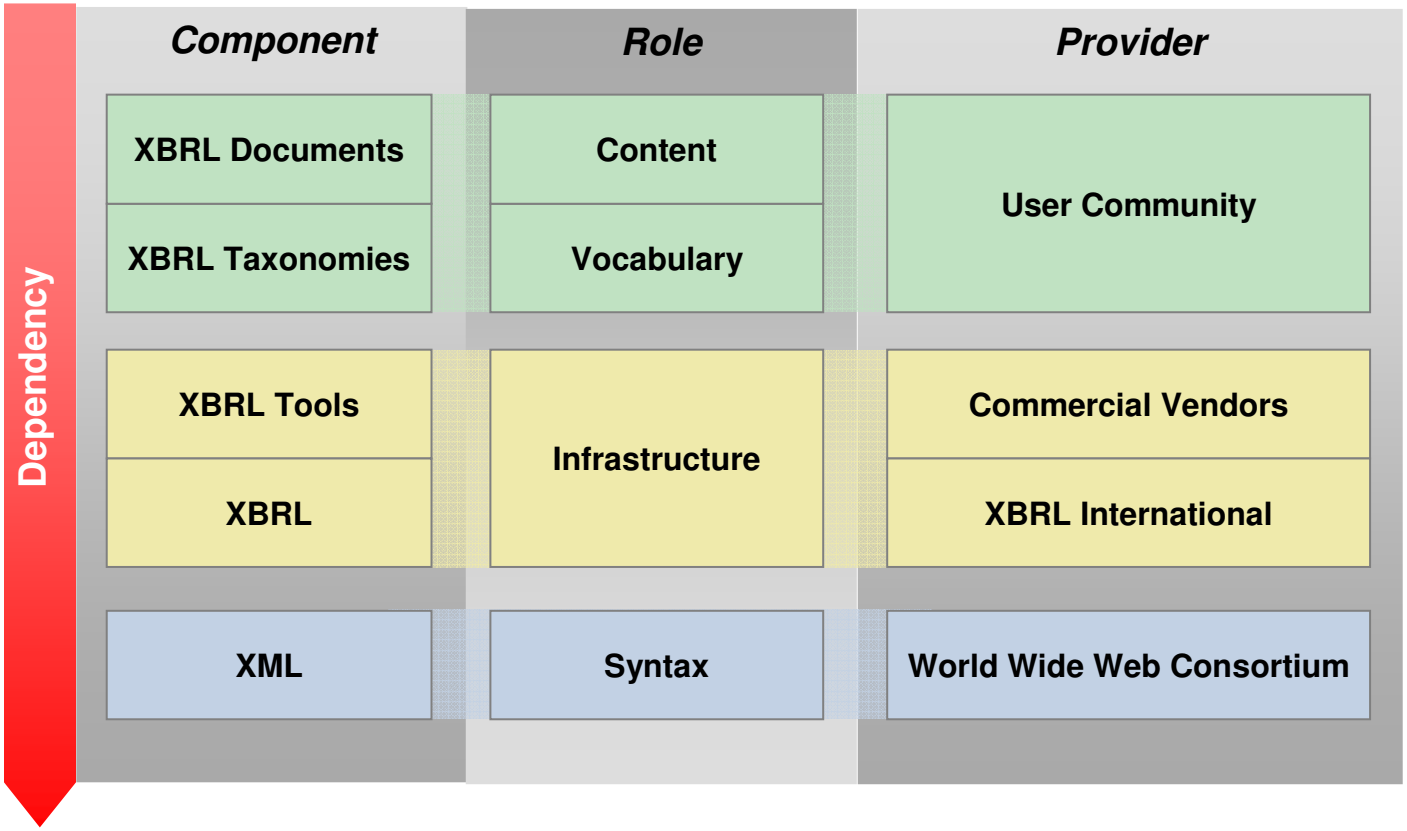


XBRL

- Visualization
- Identification
- Meaning
  - Human and Computer
- Validation
- Controlled Extension
- Automated Manipulation
- . . .

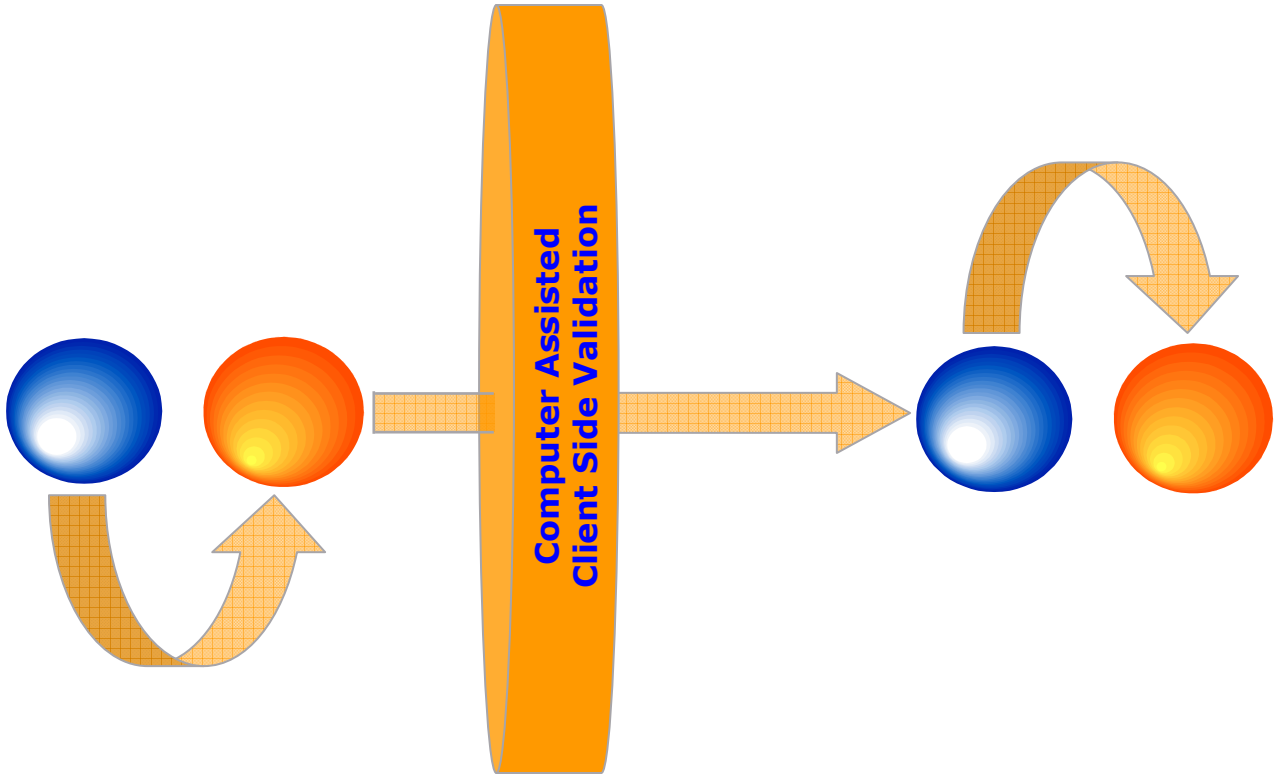
CGI

XBRL Components

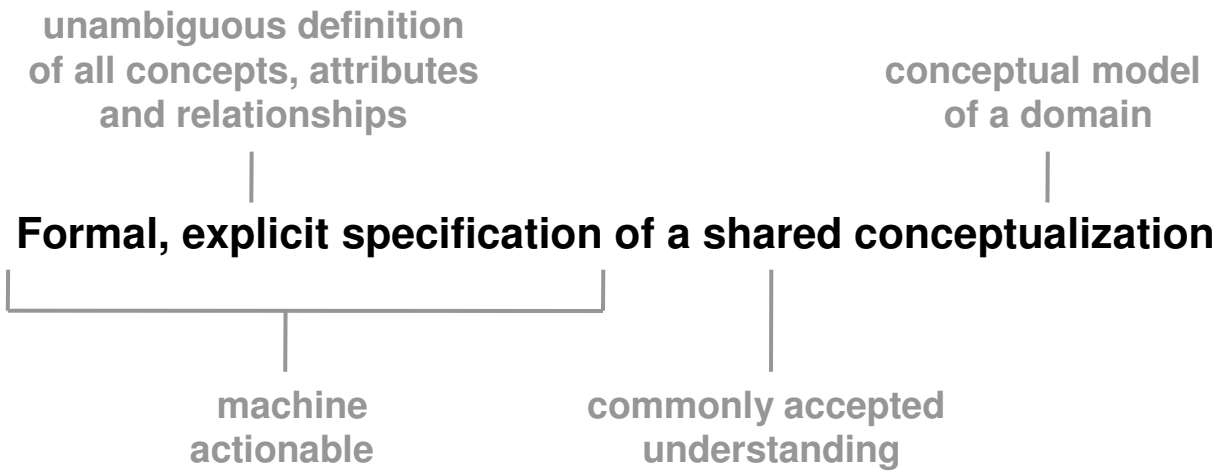


Computer Assisted Client Side Validation

- Validation of information before it is sent



Taxonomy . . . as a Common Language



From 180,000 to 8,000 data elements . . .

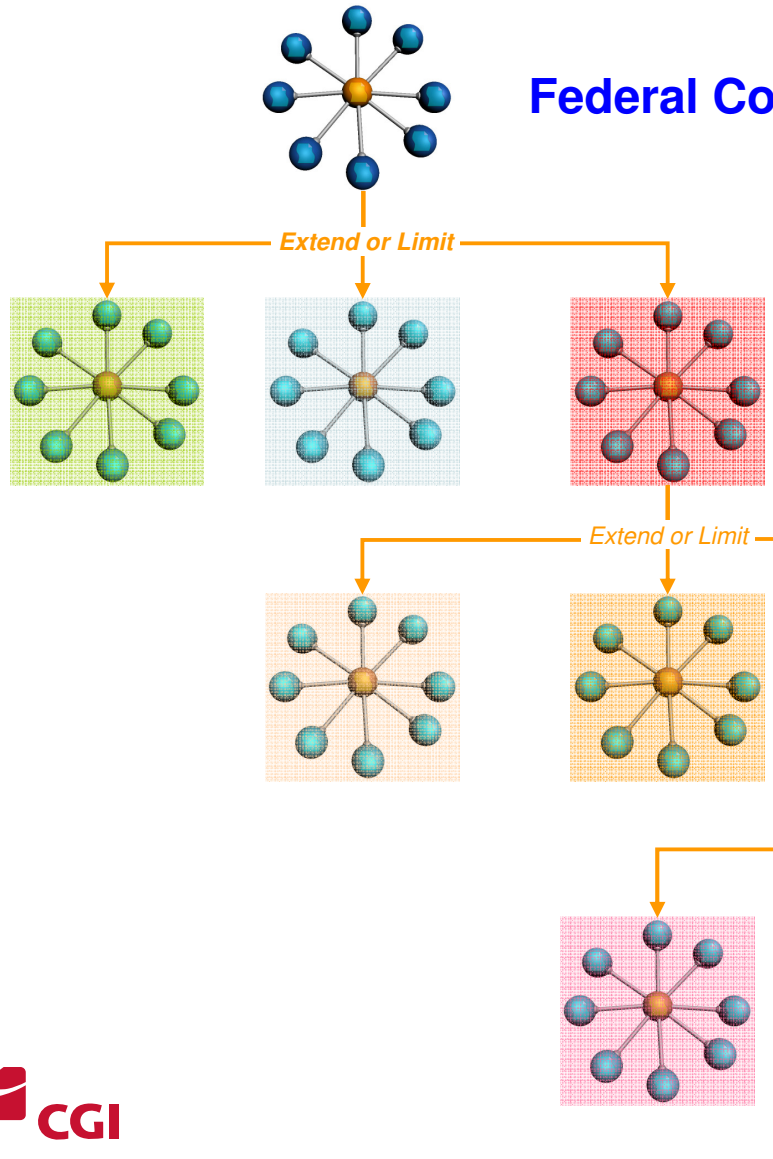
- Dutch Government

Meta Data Patterns

"The **reuse of a good solution** - a good solution to a common problem which **embodies the wisdom of experts** who have delved on the problem.

- Novices
  - Overwhelmed by options
  - Fall back to more primitive techniques
  - Over apply or misapply those few advanced techniques understood
- Experts
  - Do not solve problems from scratch
  - Reuse good solutions
  - Base new designs on prior experiences

Controllable extensibility in an Open Standard



An Extensible Information Architecture . . .

. . . . .

Separation of Concerns

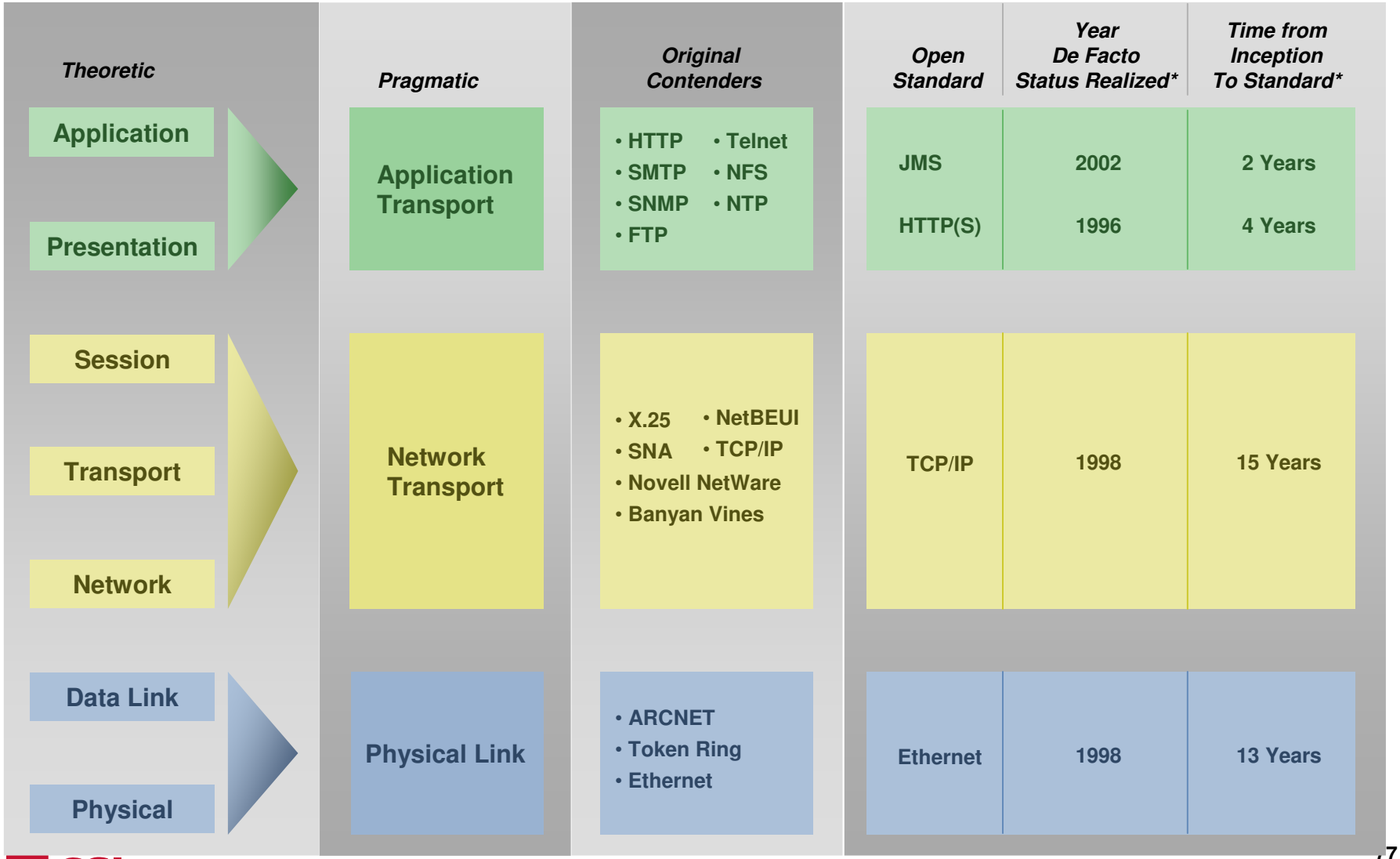
Consider what is characteristic of all intelligent thinking. It is that one is willing to study, in depth, one aspect of a subject matter for the sake of ***discerning its consistency and merit, its qualities and characteristics - all the time knowing that it is in isolation of its others aspects****. This notion of a 'separation of concerns', is an effective technique for ordering one's thoughts and discerning the essence of a thing. Focusing upon one aspect does not mean ignoring the others, rather it is just doing justice to the perspective of the one, by temporarily considering the others irrelevant.*[1]

- Edsger W. Dijkstra

[1] Extrapolated from On the role of Scientific Thought written by Dutch Computer Scientist and Turing Award winner Edsger W. Dijkstra

Conceived in the 1970's as a layered computer communication model only a subset of the whole OSI model is used today.

OSI Communication Stack

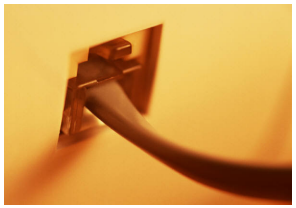


*Rough approximation

Why not just WSDL/SOAP?

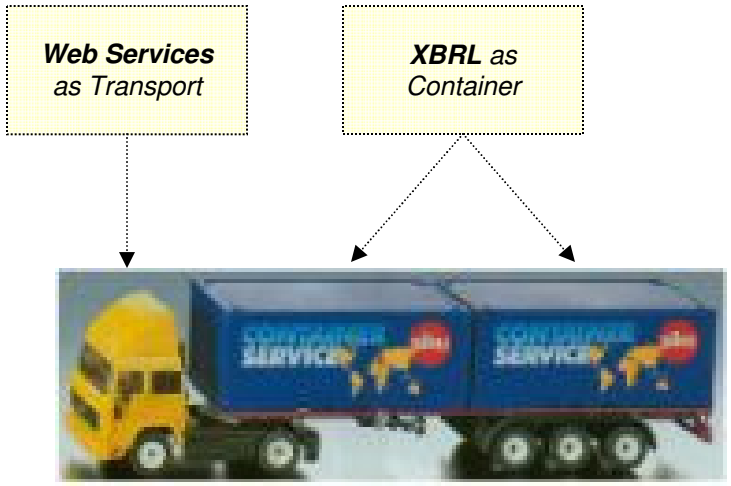
- Taking SOA where it needs to go:
  - **Reuse of Technical Elements** (avoiding the high cost of integration):
    - Instead of using an **Application Programming Interface (API)**
    - Using a **Business Transaction Interface (BTI)**
  - **Semantic Capabilities** - The missing link in the web services stack
    - **WSDL defines only a primitive (limited) contract**
      - Misses much of the business value of the transaction
    - **WSDL is too technical for business issues**
      - Should be treated as the truck and not handle the contents of the payload it carries

Containerizing Information . . .

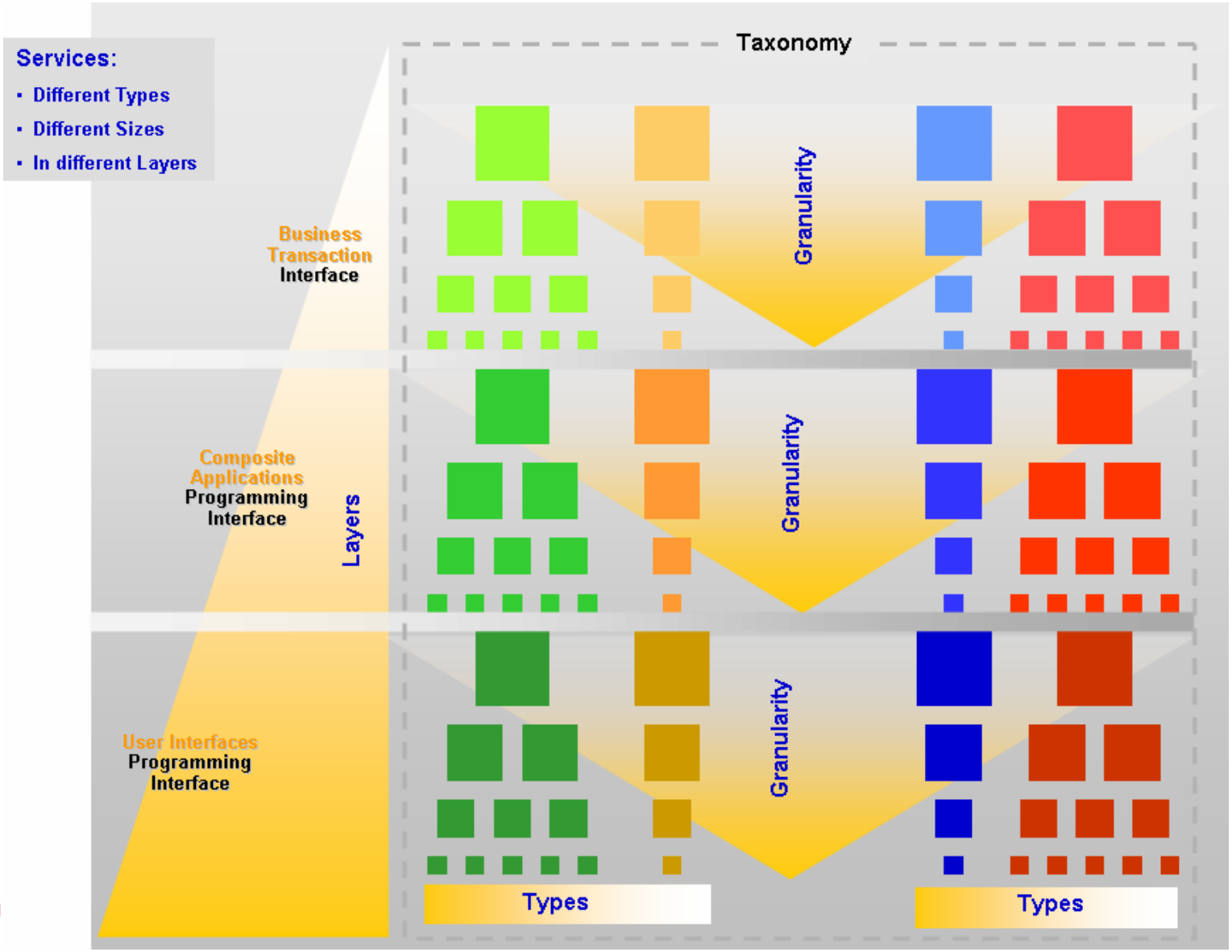


The value an Information Supply Chain is tied to the **level** of participation which in turn is tied to the **ease** of participation.

Could it be made to be as easy as plugging into a jack . . ?



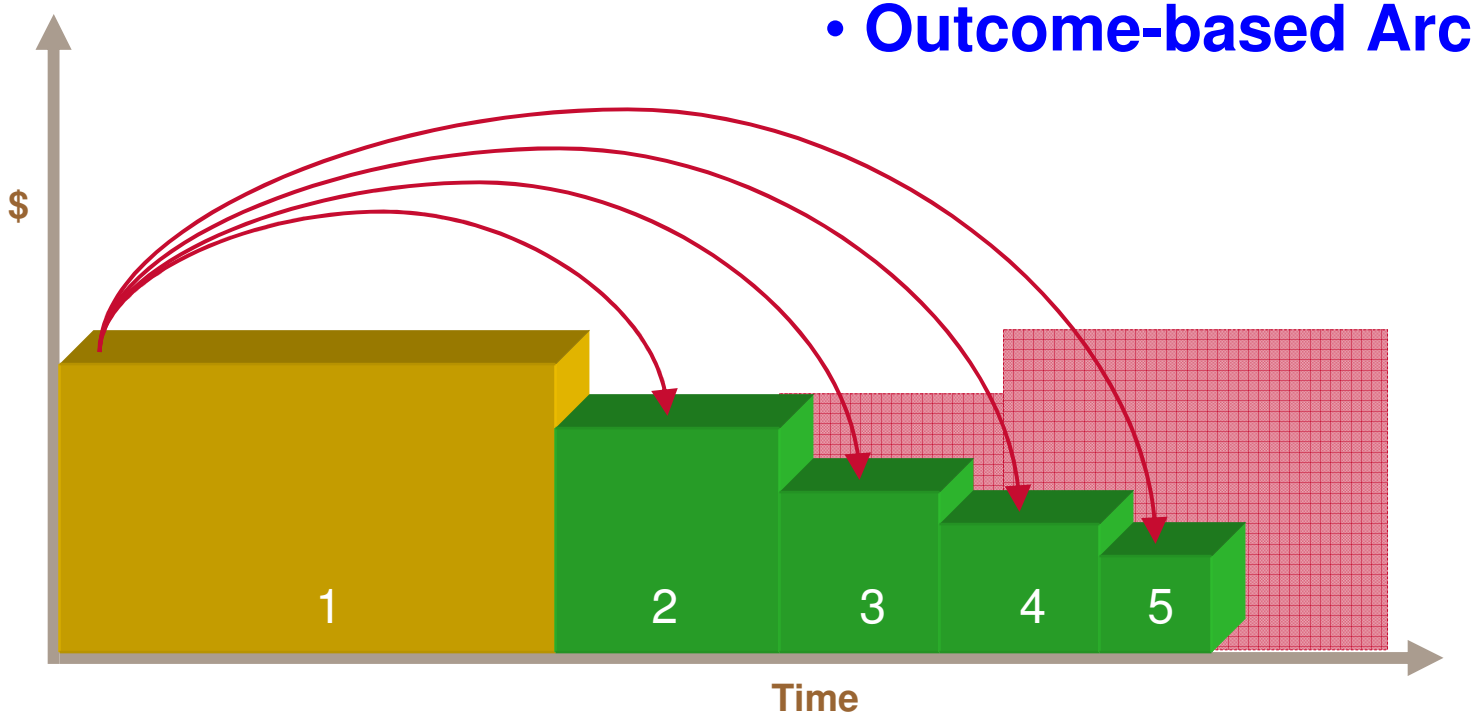
Separating Concerns



How far out . . ?

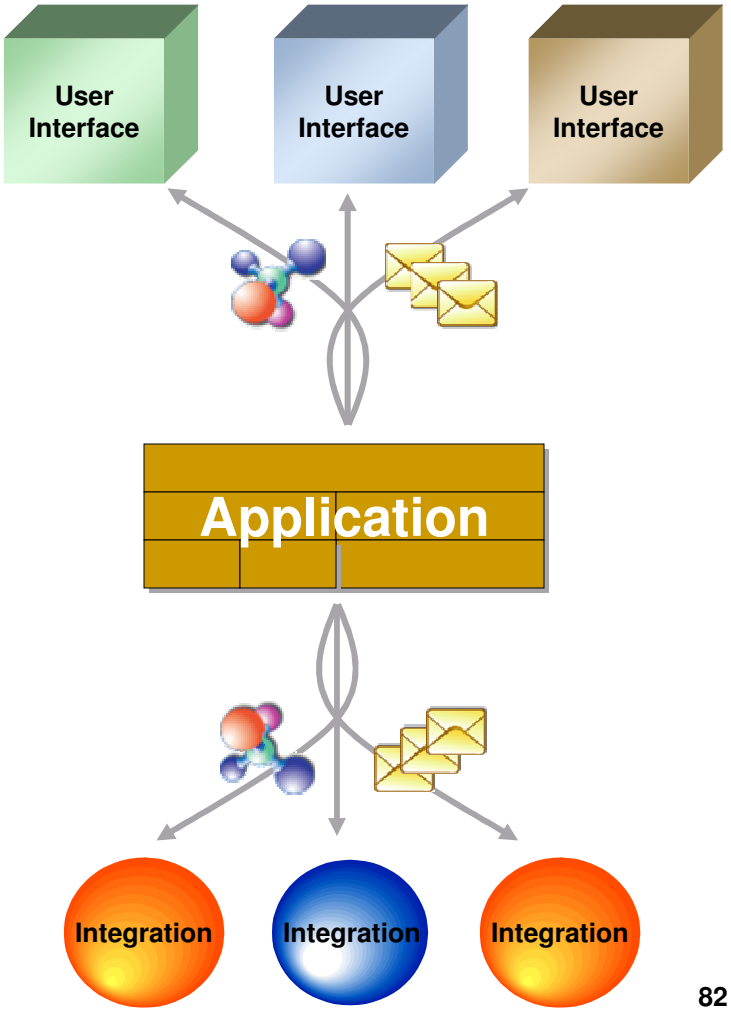
Developing **architectures for usability** impervious to both the evolution of technology and requirements . . .

- **Resilient Architectures**
- **Outcome-based Architectures**



A Broader Application of Meta data

- **Meta Driven Architecture**
  - One Application to N User interfaces
    - Optimized for power users
    - Optimized for decision makers
    - Optimized for the handicapped
  - One Application to N Integrations
    - Greater Diversity
    - Cheaper Integration



CGI

Prototype

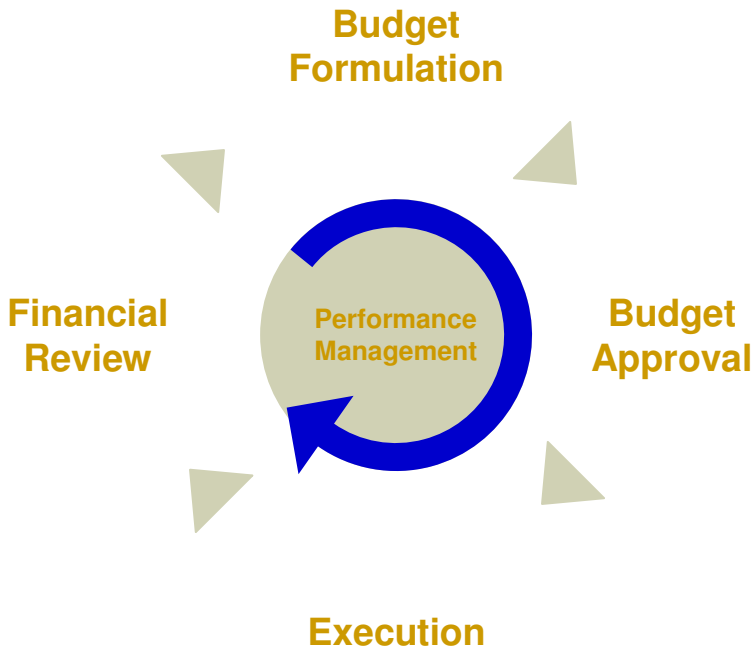
ISC Project Objectives

- **Practical Focus**
  - Choose a Use Case based on a prominent agency challenge
    - Provides a requirement that multiple Agencies will need to meet which showcases an ISC
- **Standards Based**
  - Design Standards Based Tool Kit around this Use Case
    - Not Locked into specific platform
    - More easily applied across multiple Agencies
- **Open Source**
  - Develop using Open Source applications and software
    - Allows Government to better leverage limited resources
- **Develop Freely Available Tool Kit**
  - Provides something to evaluate and build on

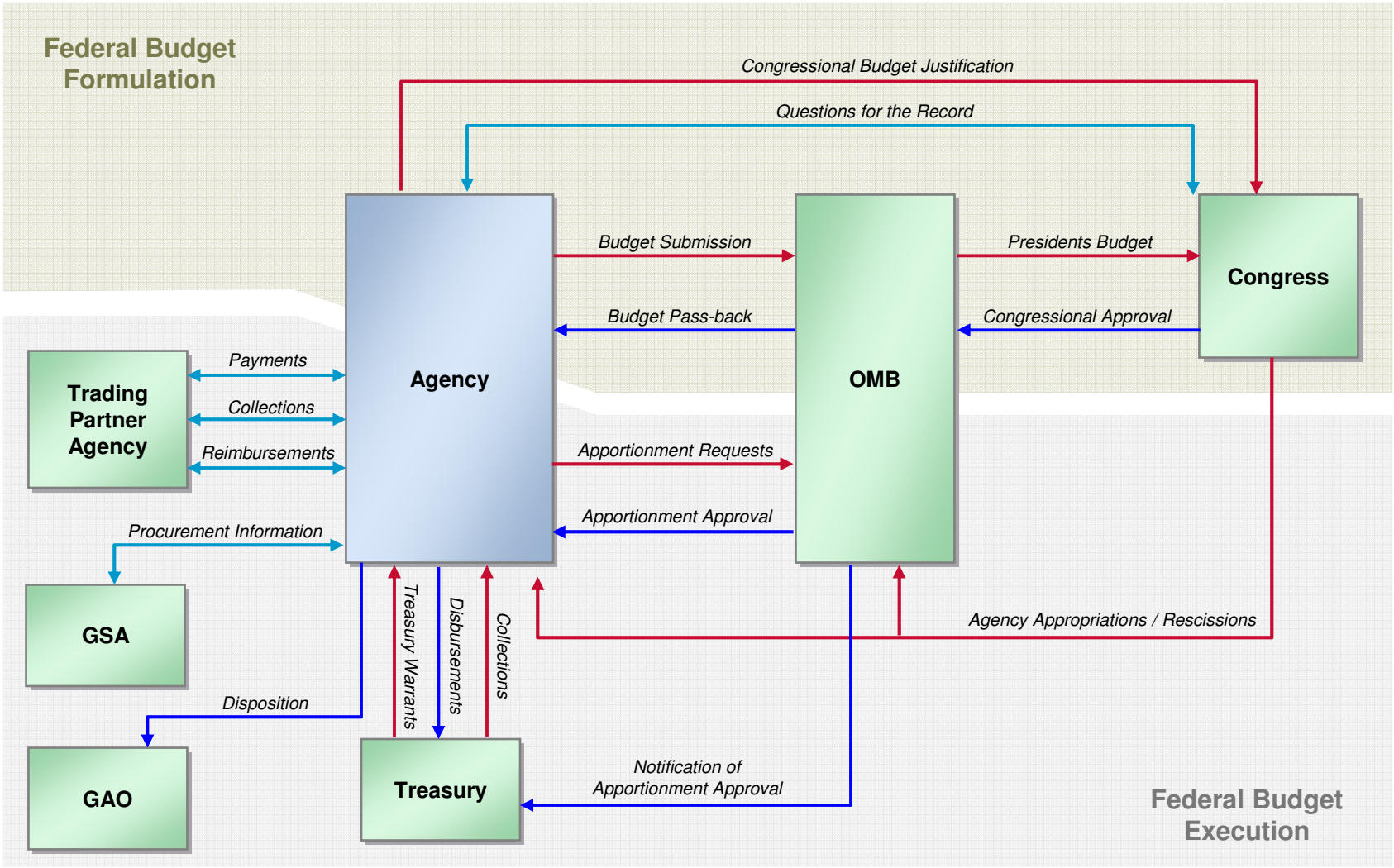
CGI

Prototype Use Case

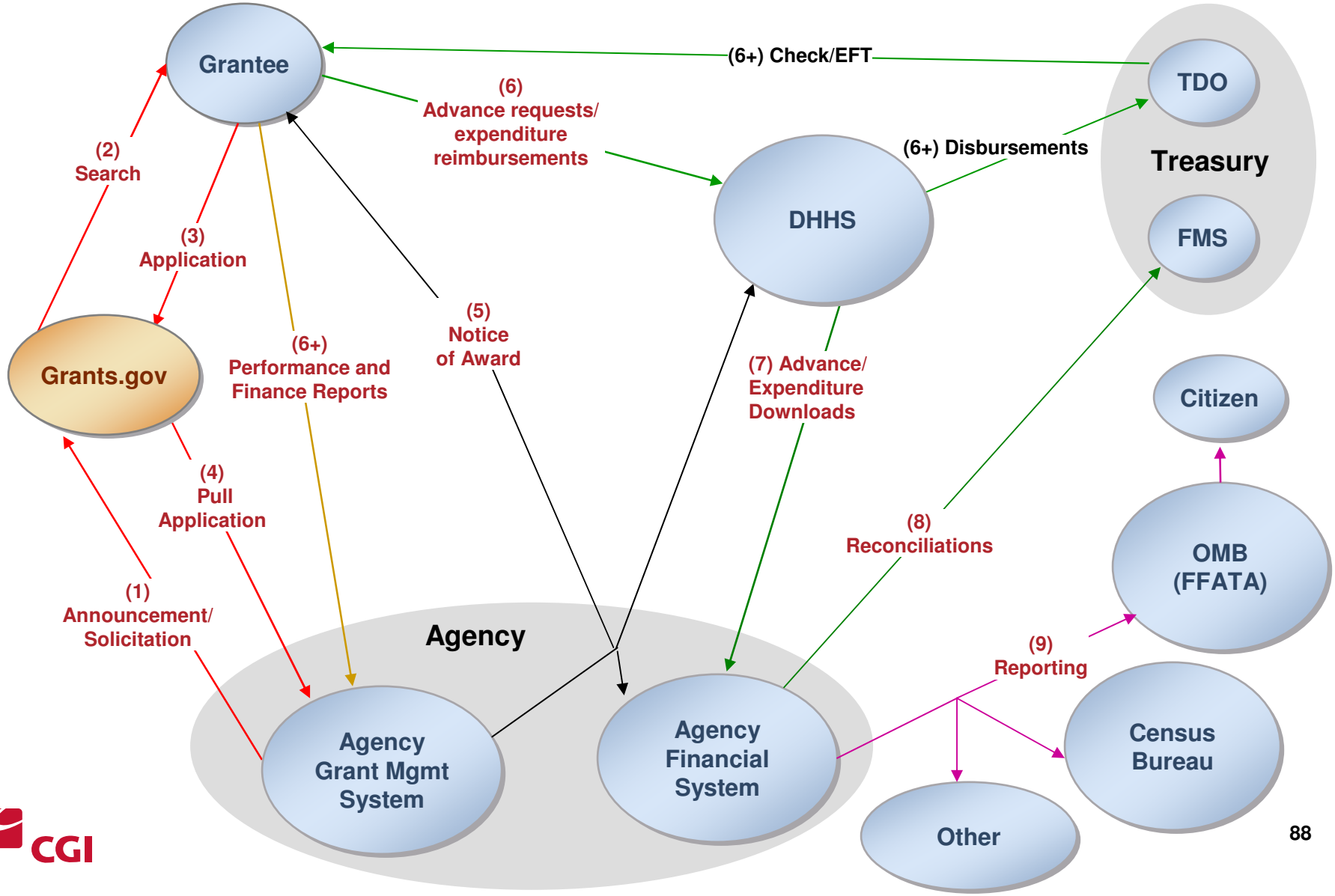
Federal Financial Management Lifecycle . . .



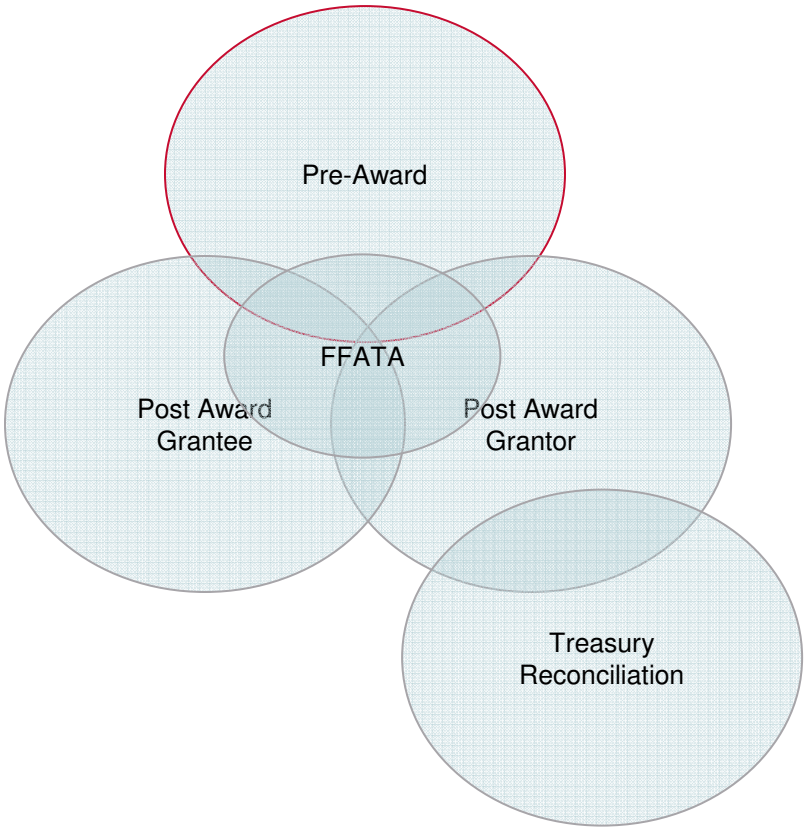
Federal Financial Management



Grants Information Supply Chain



Abstracted Grants Taxonomy . . .



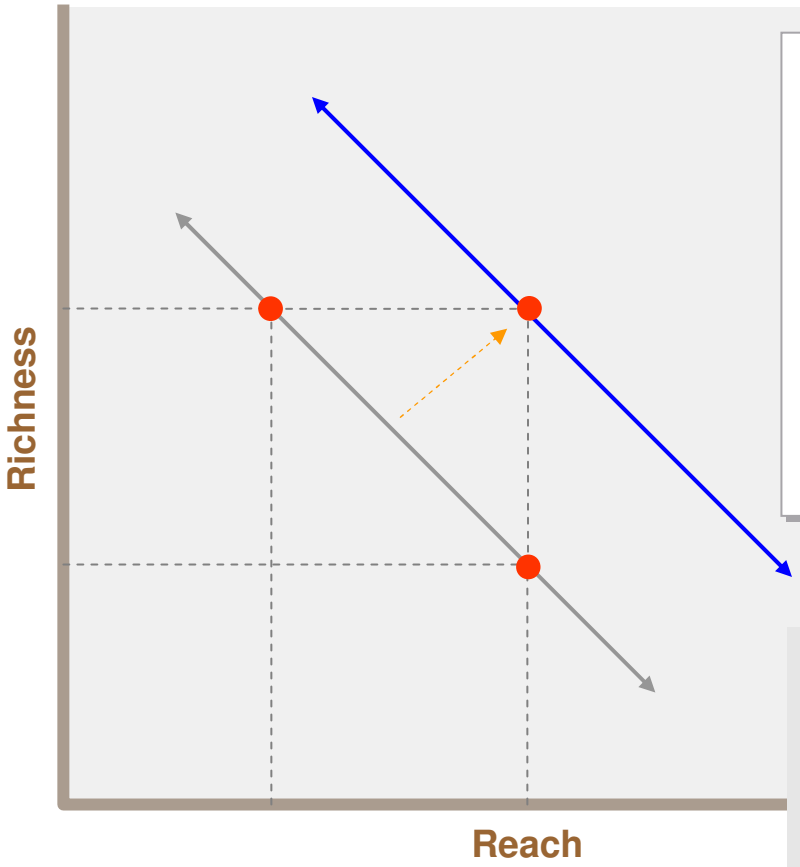
Prototype Architecture

Core Concepts of an ISC . . .

1. **Identity**
   - No longer handicapping computers - a UPC code for Data
2. **Meta Data**
   - Why not Reuse of:
   - Information?
   - Information about information (Meta Data)?
3. **Self Directed Messages**
   - Asynchronous
   - WS - Address
4. **Separation of Concerns**
   - Using a Technology for only a single purpose at which it excels
5. **Communication**
   - Scope of communication as proportionate to its value (Metcalfe's Law)
   - Economics of Information . . . The rising ceiling on Richness vs Reach
6. **Extensibility**
   - Having something that is common and yet extensible as the fundamental premise of a Standard

CGI

Raising the Ceiling . . .



Richness refers to the quality of information, as defined by the user and reflected in characteristics of the information such as accuracy, timeliness, etc.

Reach refers to the number of people who participate in the sharing of that information.

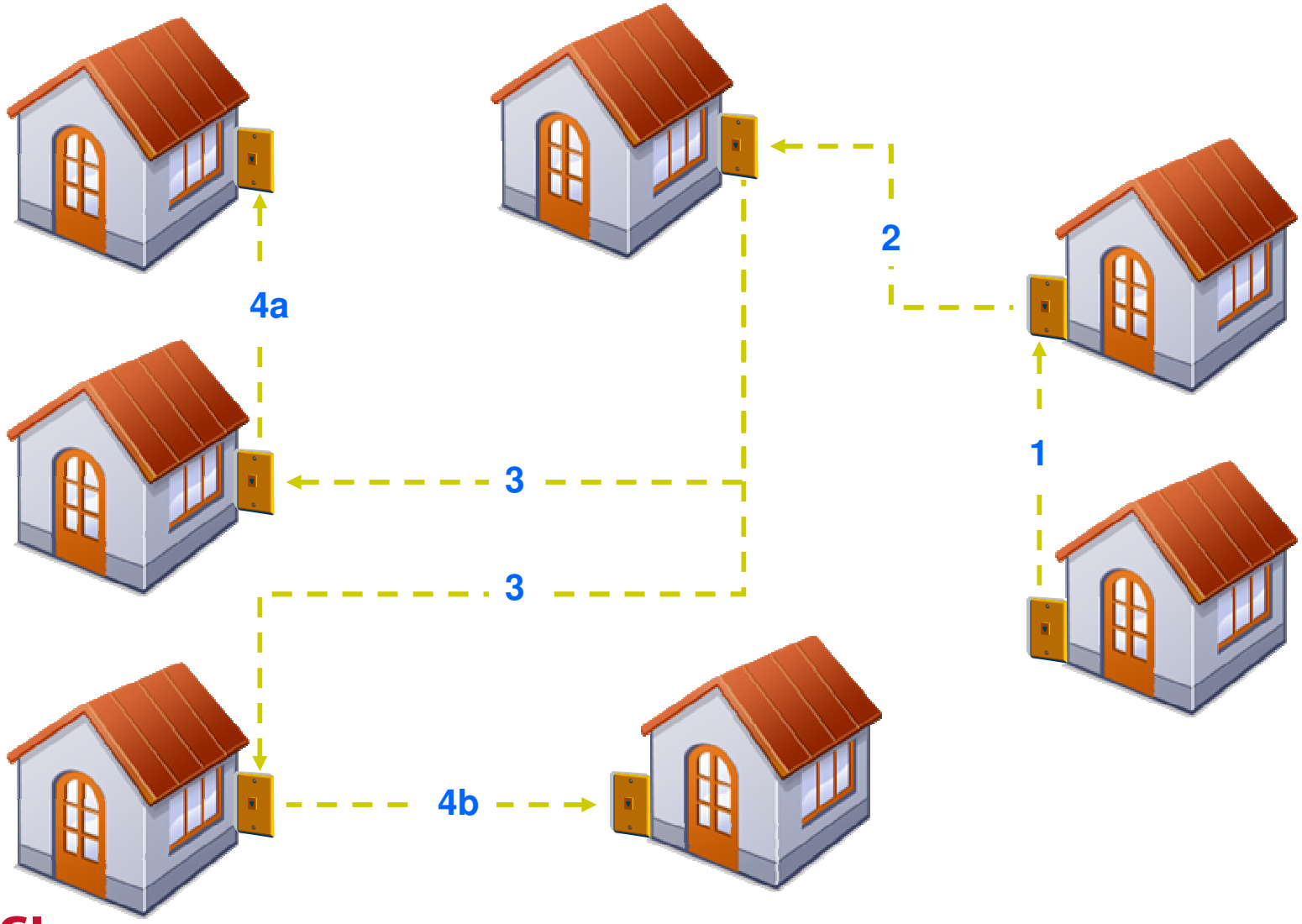
By unbundling information from its physical carrier, a greater level of richness and reach can be concurrently achieved, raising the ceiling on the Economics of Information.

CGI

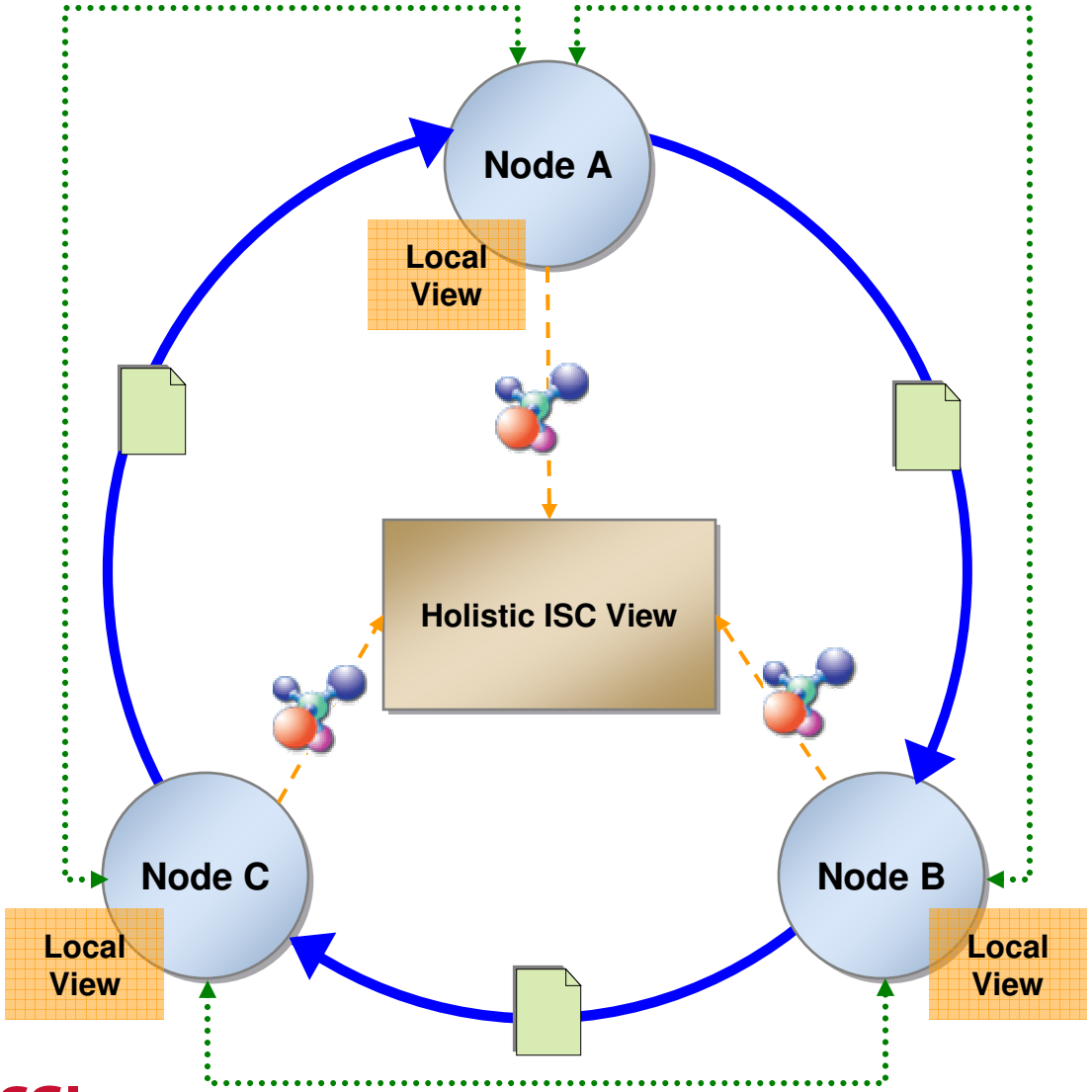
Technology Stack

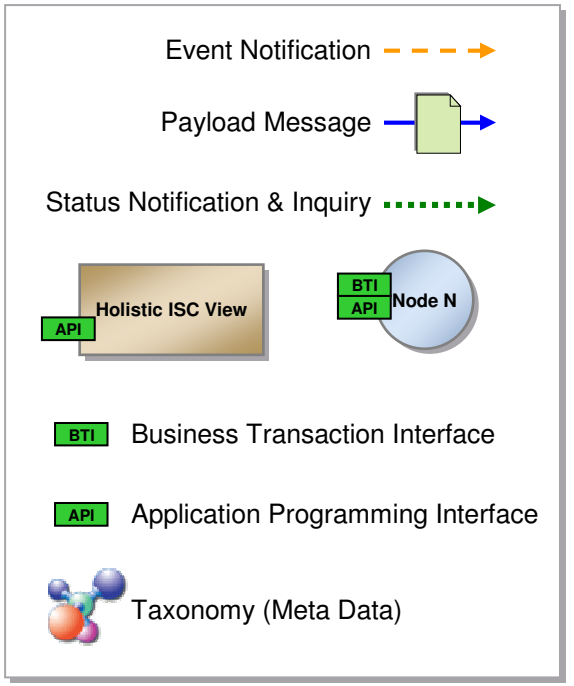
Layer	Standard	Solution
Process UI	*Meta Data*	Adobe Flex
Semantic UI	XBRL	UB Matrix
Semantic Taxonomy Editor	XBRL	UB Matrix
Semantic Engine	XBRL	UB Matrix
Service Orchestration	BPEL	Intalio
ESB	SOAP/WSDL	FUSE
SOAP Engine	SOAP/WSDL	FUSE
Web Server	Java	Apache Tomcat
Application Server	JEE	-
Object Relation Mapping	JEE	-
Database	XML	Sedna
Operating System		Windows
Hardware		Dell

Enabling an ISC . . . ‘A Black Box’

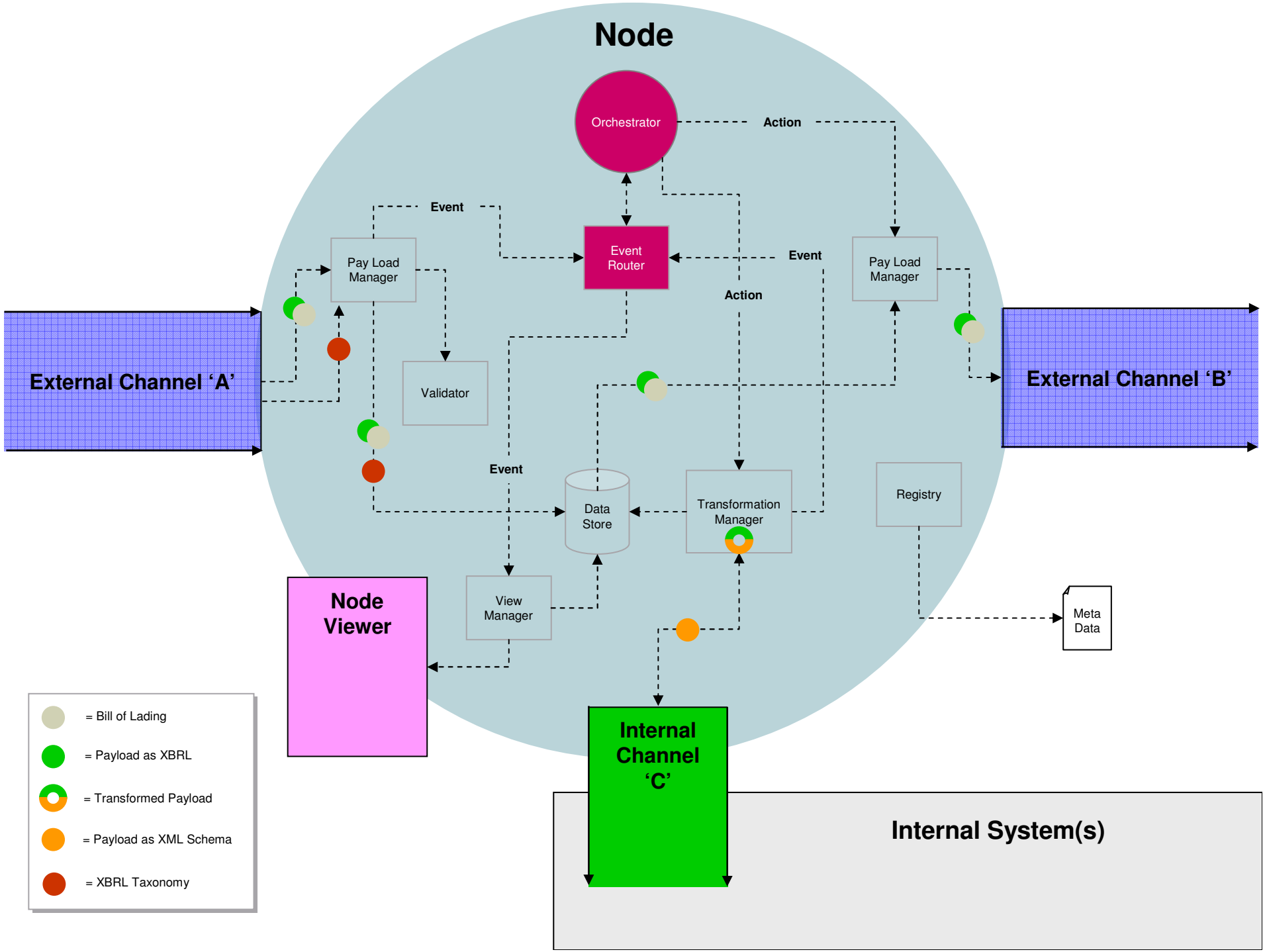


ISC Node Interactions

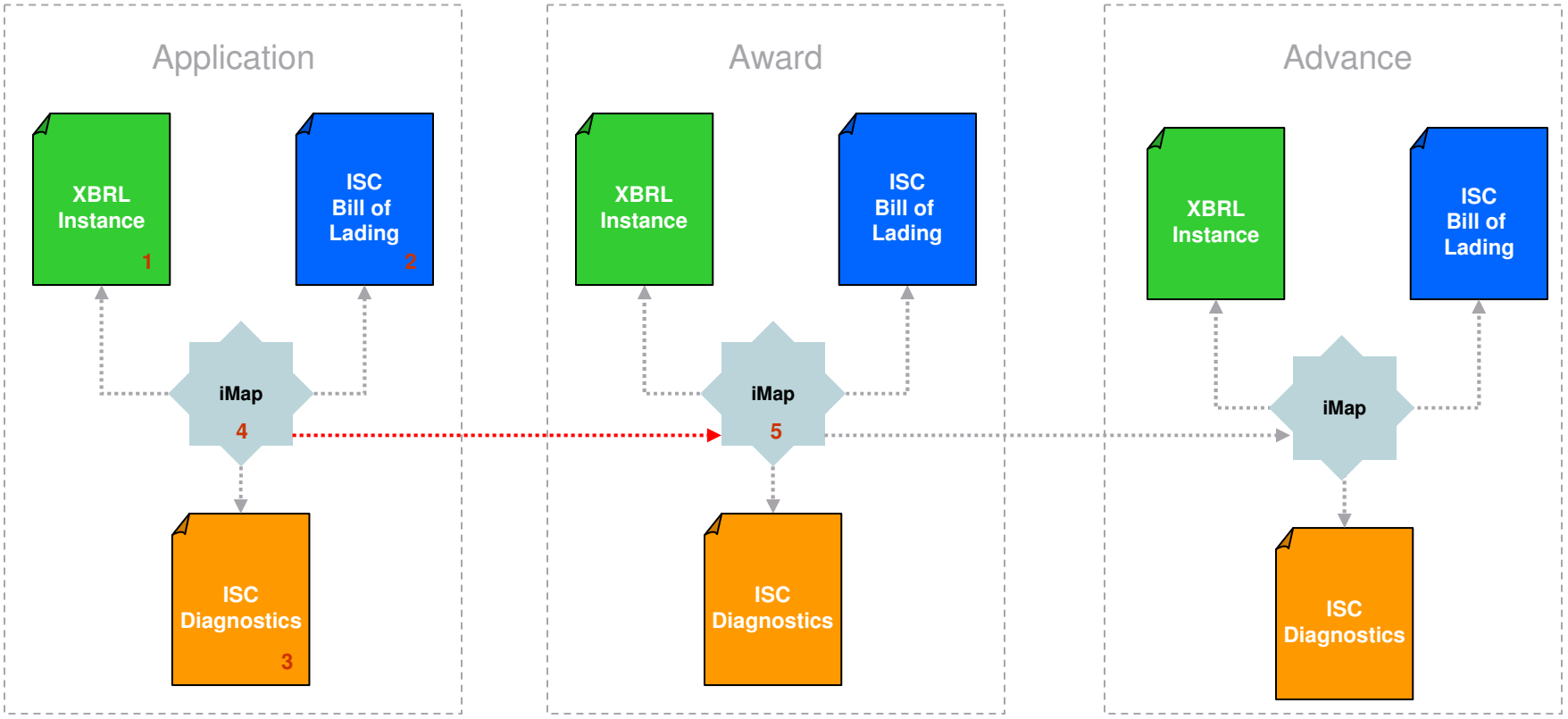




CGI

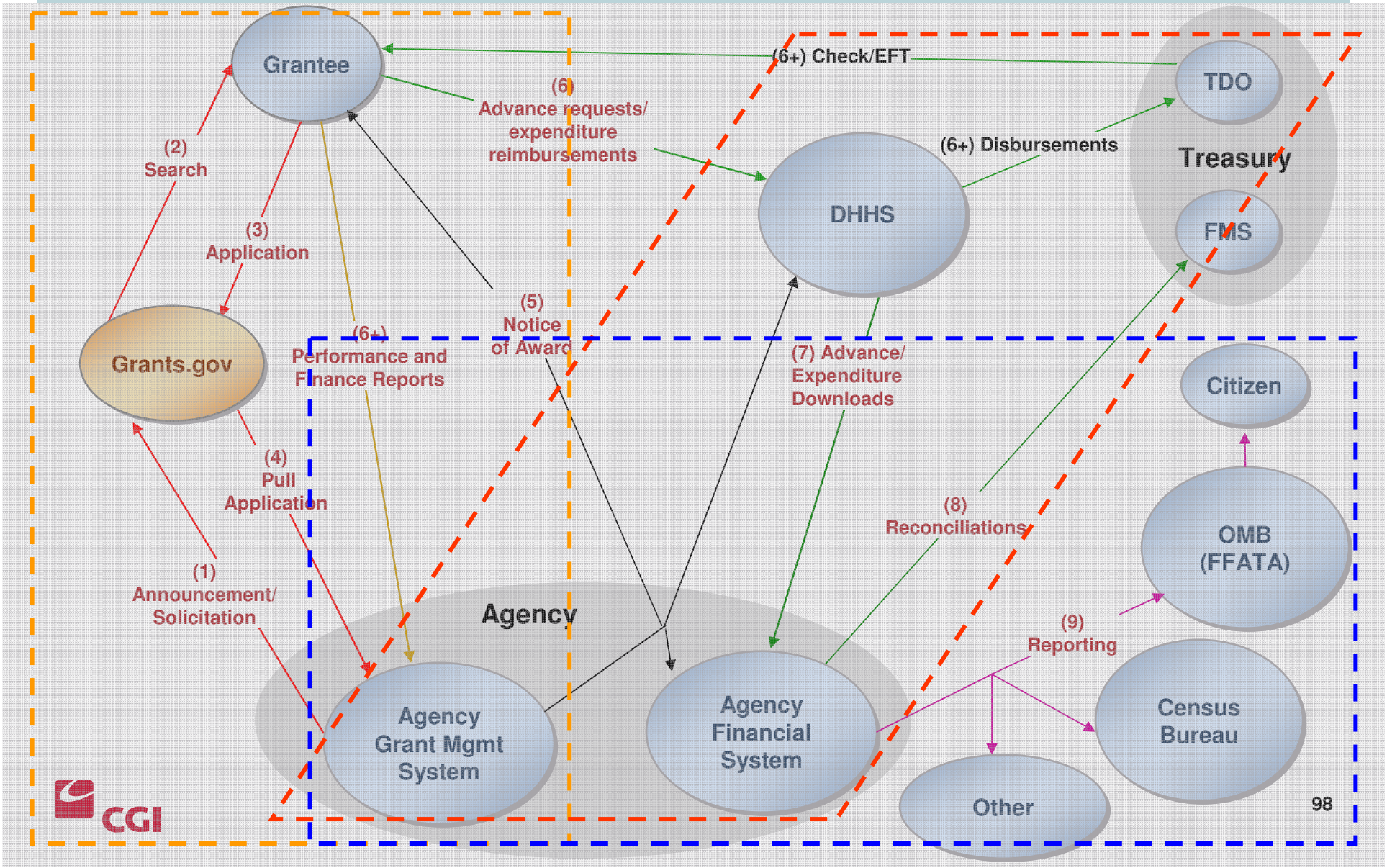

Node
Orchestrator
Action
Event
Event Router
Event
Action
Pay Load Manager
Pay Load Manager
External Channel 'A'
External Channel 'B'
Validator
Event
Data Store
Transformation Manager
Registry
Node Viewer
View Manager
Meta Data
Internal Channel 'C'
Internal System(s)
= Bill of Lading
= Payload as XBRL
= Transformed Payload
= Payload as XML Schema
= XBRL Taxonomy

Logically Relating Document Sets

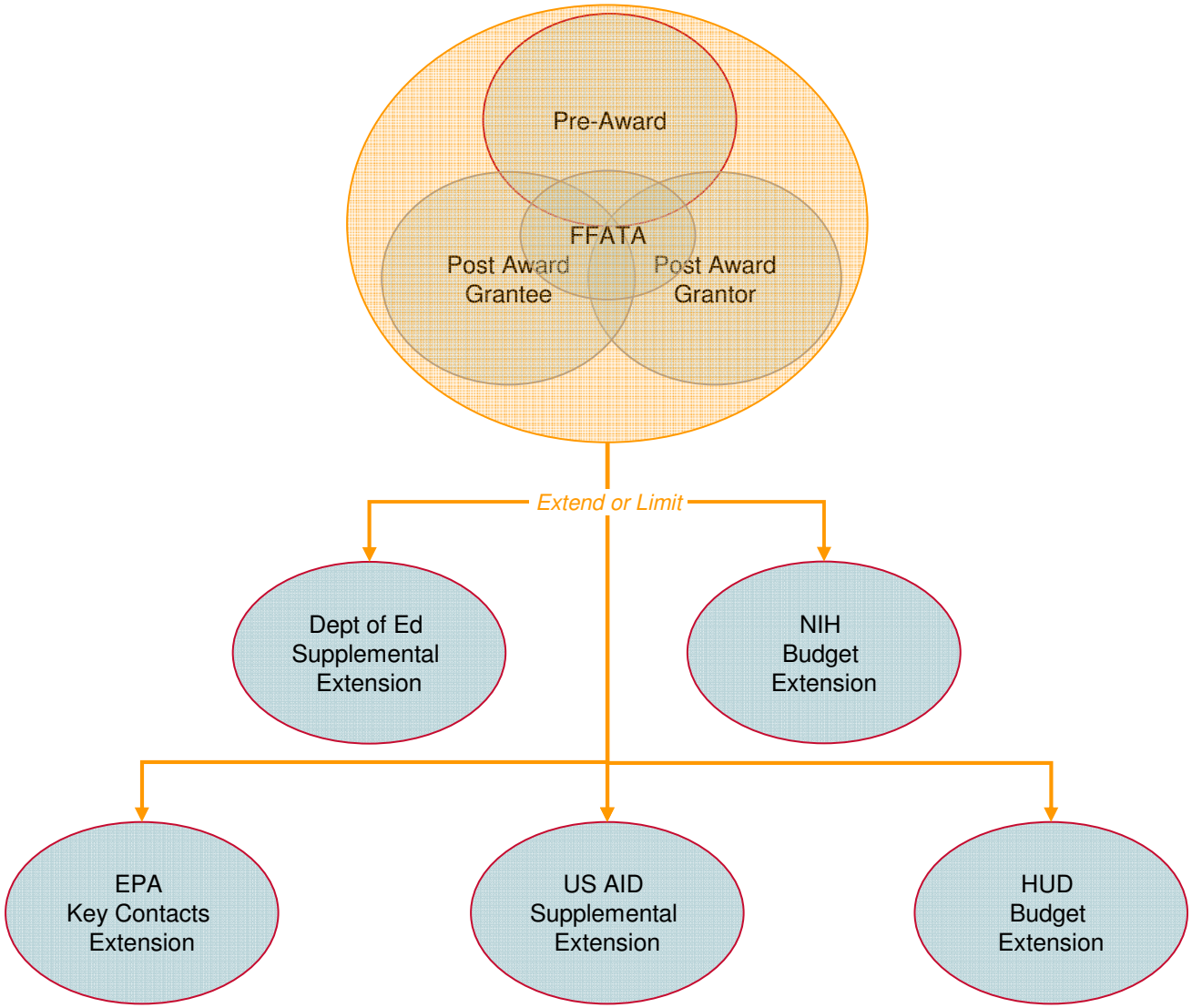


CGI

Leveraging Meta Data across the Information Supply Chain . . .

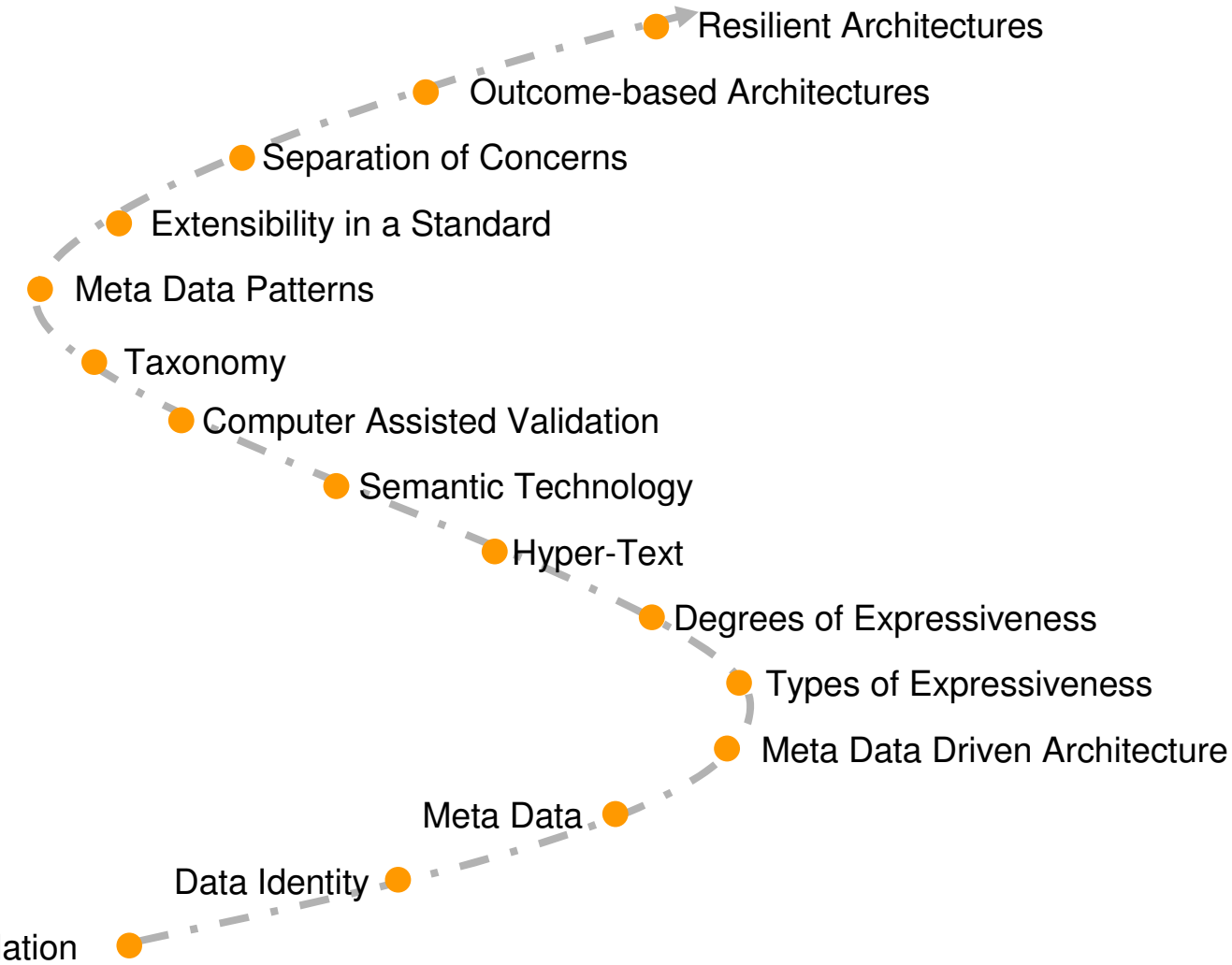


Extended Grants Taxonomy



Business Perspective

A Progression of Concepts . . . Business Architecture



Proper Decisions

> **Data is the heart** of any business. Without good data turned into information, management can't make the **proper decisions**.[26]

- Neal J. Hannon, Strategic Finance Magazine September 2005

Perfectly Increasing Returns

> Information has "***perfectly***" increasing returns; spend the money to learn something once, and that knowledge can be reused at zero additional cost forever; double the number of uses and the cost per use halves. [1]

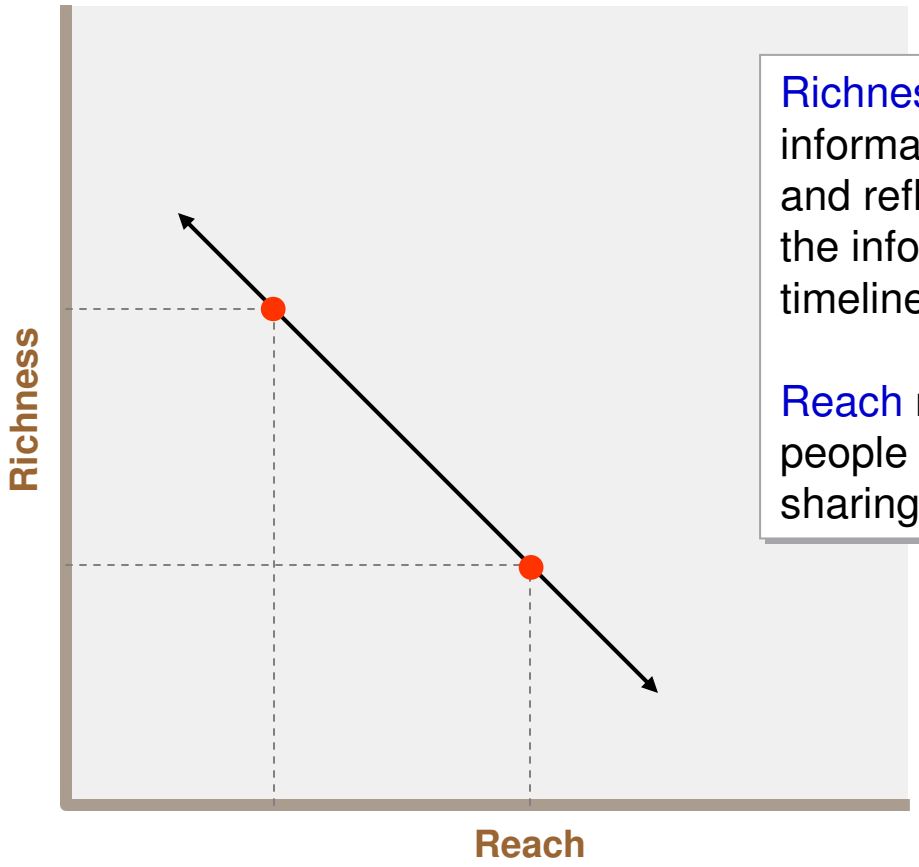
[1] Philip Evans and Thomas S. Wurster, Blown to Bits: How the New Economics of Information Transforms Strategy, Harvard Business School Press (October 1999), p15

CGI

Visibility

The ease with which financial information can be **accessed and manipulated** for analytical purposes is **directly tied to** the degree of **visibility** into an entity's performance.

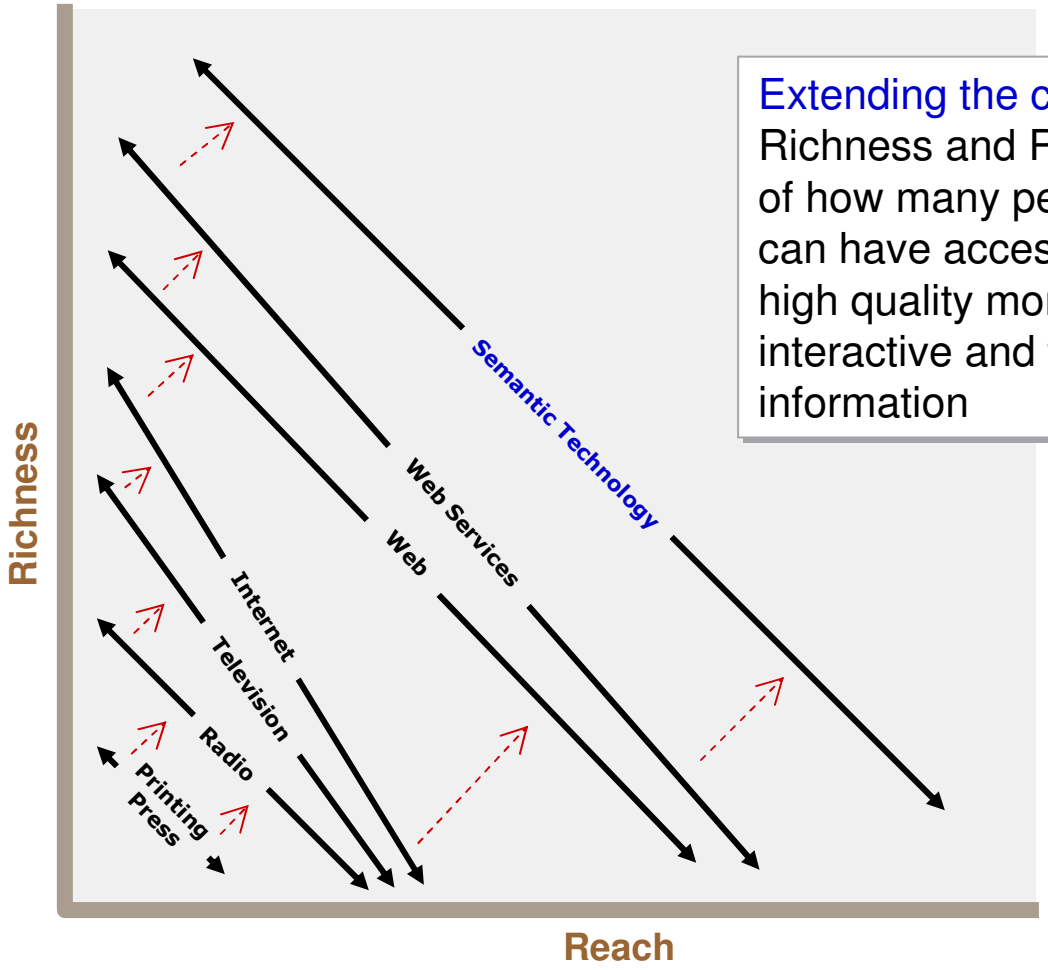
Economics of Information . . .



Richness refers to the quality of information, as defined by the user and reflected in characteristics of the information such as accuracy, timeliness, etc.

Reach refers to the number of people who participate in the sharing of that information.

Extending the Ceiling . . .



Extending the ceiling of Richness and Reach – of how many people can have access to high quality more interactive and timely information

Metcalf's Law . . . The Network Effect

- Metcalfe's law States that the value of a telecommunications network is proportional to the square of the number of users of the system (n2).
- First formulated by Robert Metcalfe in regard to Ethernet, Metcalfe's law explains many of the network effects of communication technologies and networks such as the Internet and World Wide Web.
- The law has often been illustrated using the example of fax machines: A single fax machine is useless, but the value of every fax machine increases with the total number of fax machines in the network, because the total number of people with whom each user may send and receive documents increases.

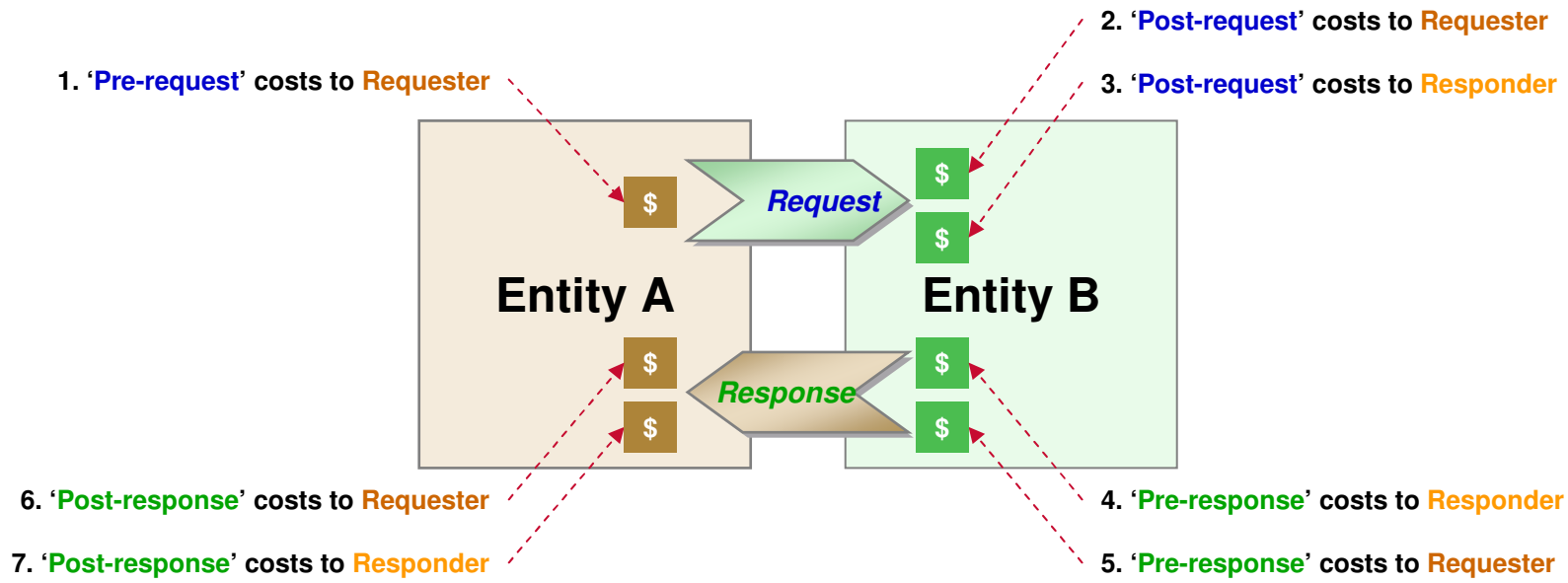
Network effect

From Wikipedia, the free encyclopedia

A **network effect** is a characteristic that causes a good or service to have a value to a potential customer which depends on the number of other customers who own the good or are users of the service. In other words, the number of prior adopters is a term in the value available to the next adopter.

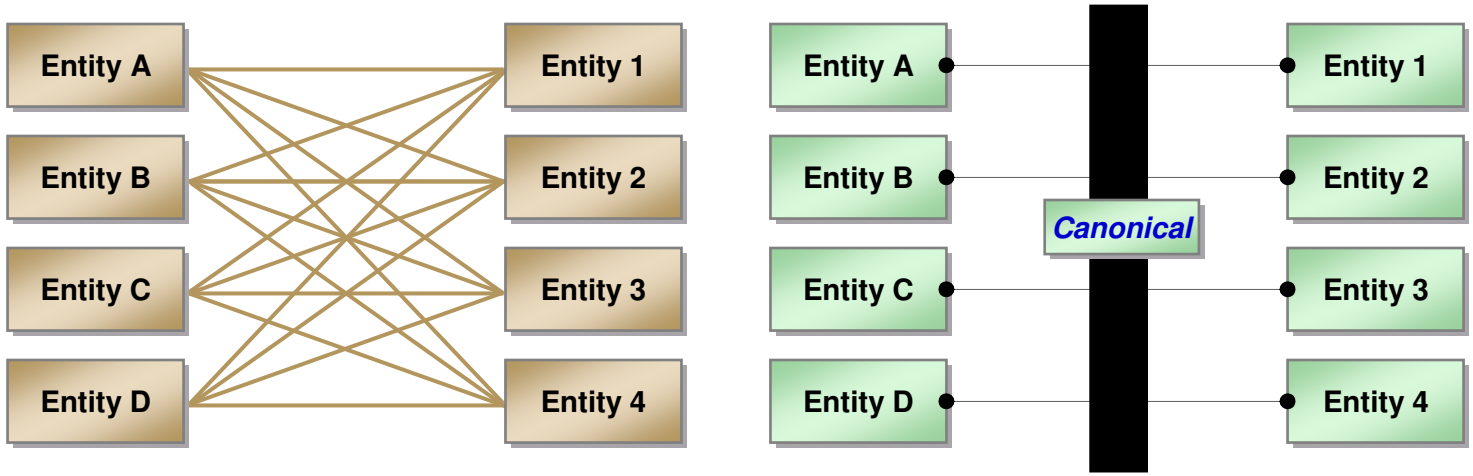
One consequence of a network effect is that the purchase of a good by one individual indirectly benefits others who own the good — for example by purchasing a telephone a person makes other telephones more useful. This type of side-effect in a transaction is known as an externality in economics, and externalities arising from network effects are known as **network externalities**. The resulting bandwagon effect is an example of a positive feedback loop.

The Unnecessary Cost of Information Sharing



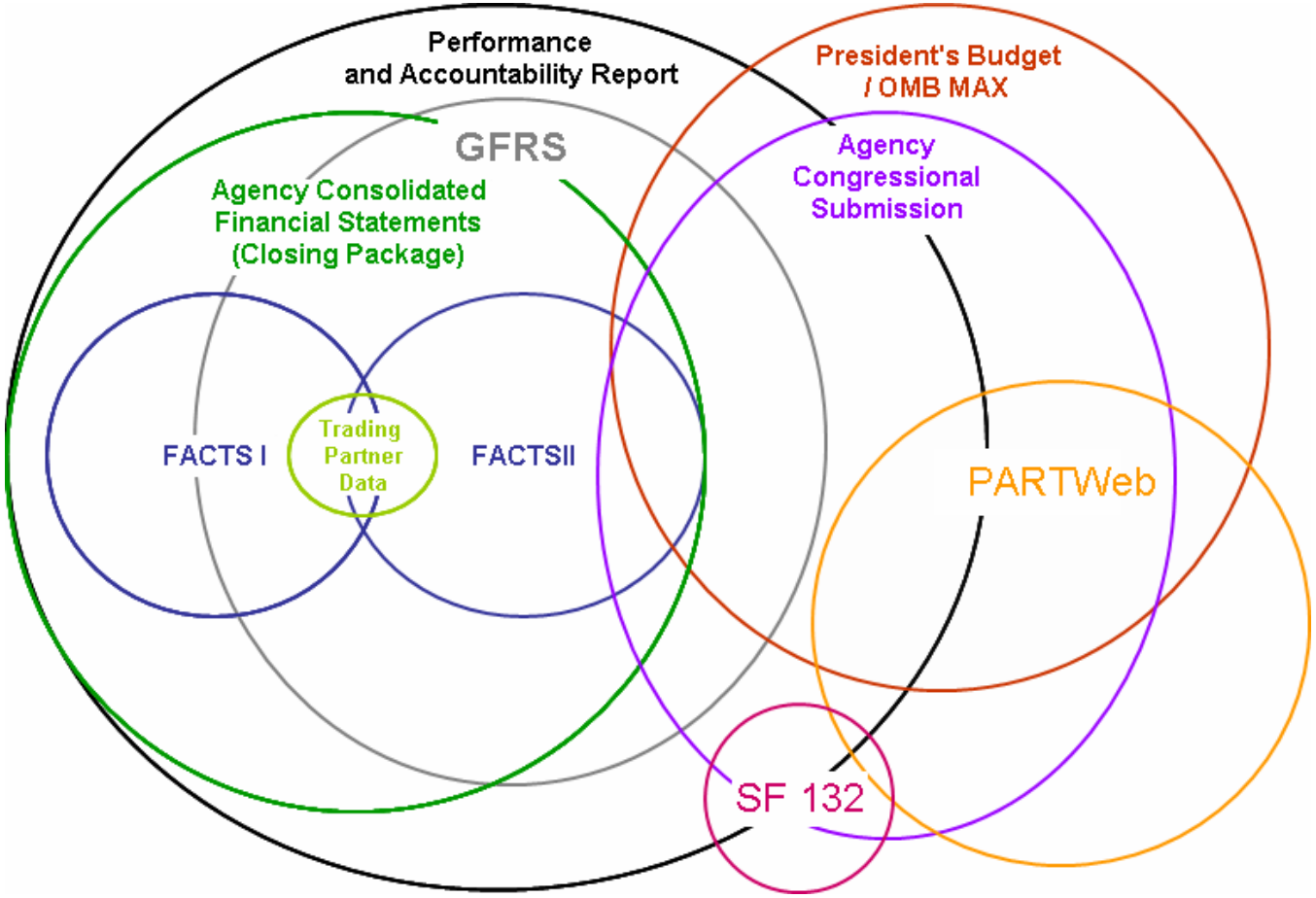
#	Role Bearing Cost	Timing of Cost	Traditional Reporting Cost
1	Requester	Pre-request	Gathering Requirements / Documenting Requirements / Formatting Requirements / Transmitting Requirements
2	Requester	Post-request	Answering questions/ giving clarifications / making corrections to the request
3	Responder	Post-Request	Interpreting Request
4	Responder	Pre-response	Gathering / Verifying / Reconciling / Summarizing / Formatting / Request Interpretation
5	Requester	Pre-response	Answering questions and giving clarifications (given actual responder may not have been recipient of request)
6	Requester	Post-response	Interpreting Response / Verifying / Reconciling / Consolidating / Summarizing / Formatting
7	Responder	Post-response	Answering questions and giving clarifications / Researching Inquiry / Making Corrections / Reconciling changes with original response

Imposed Cost

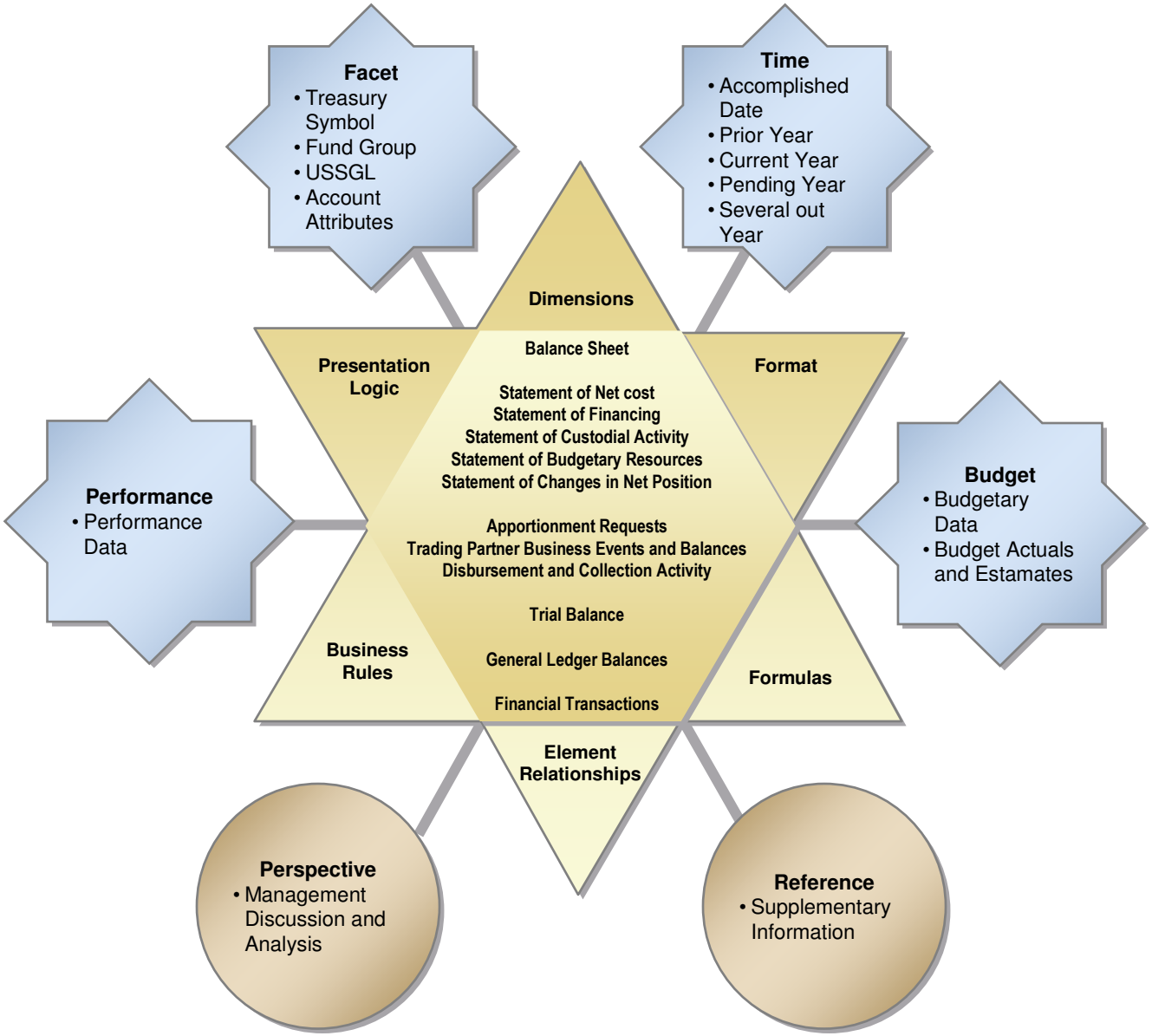


The traditional view of financial reporting as an isolated unique need between only two entities has **imposed** a considerable cost upon the federal government.

Many Different Languages . . .



One Common But Extensible Language



Benefits . . . In a flexible Information Architecture

Benefit	Description
Accuracy	A taxonomy specifies the meaning and rules of valid data, while automated tools can insure the compliance with the taxonomy. Data is validated pre-submission.
Consistency	A taxonomy acts as a dictionary, providing an explicit definition for each data element that can easily be shared to assure consistent interpretation. The taxonomy enables groups or communities to represent and share a common set of terminology, in an open, transparent and efficient manner.
Efficiency	The combination of taxonomies, XML based documents, and automated tools enables the automated processing of business information and eliminates the manual processes of validation, re-entry, and comparison.
Reuse	A form of efficiency, but worth noting separately when contrasting to the historical notion of reporting. By marrying an XML document with a taxonomy, XBRL is able to provide information in a format optimized for reuse - letting format, level of detail, and presentation be the choice of the end user rather than the information provider.
Flexibility	Unlike other XML standards, XBRL was architected for agility in many different contexts: A) letting end users determine how they wish to view and manipulate information, B) allowing information providers to extend taxonomies for new information exchanges without undermining existing taxonomies or compatibility with existing tools, or C) enabling taxonomy updates to be applied rapidly and without programmatic changes.
Traceability	The fact that information is provided with a mapping to a taxonomy allows for greater traceability in determining both: A) From where it was derived and B) To what it relates. No longer does data have to be stripped of all supporting information and become just a number.
Resiliency	Requirements and technology never stop evolving. It is critical that technology be crafted to evolve and extend with the needs of an organization in an efficient fashion. The extensive use of meta data by XBRL along with the separation of concerns allows this to happen.
Visibility	The ease with which information can be accessed and manipulated for analytical purposes defines the degree of visibility into any organization, issue, or subject of interest. XBRL, through its layered component-based architecture, can dramatically enhance the visibility into financial matters such as an organization's performance or effectiveness.

Information Opportunities with XBRL

Opportunities/Benefits	*Benefits*						
	Accuracy	Consistency	Efficiency	Reuseability	Flexibility	Traceability	Visibility
Single Taxonomy Across All Reports	✓	✓	✓	✓	✓		
Embed Detail Within Hierarchical Taxonomy	✓	✓	✓	✓	✓	✓	
Single Reporting Window - Leverage Taxonomy for Single Information Exchange		✓	✓	✓			
Computer and Human Readable Report Format			✓	✓	✓	✓	
Data Sharing Between Interested Parties		✓	✓	✓		✓	✓