World Trade Symposium 2018: Open Architecture Round Table Session

Open Markets • Open Finance • Open Architecture

Silos ⟶ Nodes

How can silos that prevent interoperability be transformed into nodes of a network?

Facilitated by Joseph Potvin, Xalgorithms Foundation
jpotvin@xalgorithms.org • +1-819-593-5983 • https://xalgorithms.org • https://internetofrules.org

Group Discussion Notes

	Expression ***How is it stated?***	Interface ***How is it accessed?***	Processor ***How does it work?***	Channel ***What pathway?***	Infrastructure ***What hardware?***
Ideas & Opportunities	*The payloads for interoperable computation should use language-independent syntax* *Examples:* *JSON* *https://www.json.org/* *Tuple programming (i.e. control tables, decision tables)* *http://wiki.c2.com/?TupleOrientedProgramming*	*Interfaces should implement standard semantic specifications* *Examples:* *Universal Business Language (UBL)* *https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ubl* *Pan European Public Procurement On-Line (PEPPOL)* *https://peppol.eu*	*The system should implement the standard non-repudiation* *Reference:* *ISO/IEC 13888-2:2010 "Information technology -- Security techniques -- Non-repudiation -- Part 2: Mechanisms using symmetric techniques"* *https://www.iso.org/standard/44736.html*	*The system design should incorporate "mult-path" routing.* *Reference:* *https://en.wikipedia.org/wiki/Multipath_routing*	*The system design should reflect 'resilience thinking'* *References:* *Smith, P. et.al. (2011). Network resilience: a systematic approach. IEEE Communications https://doi.org/10.1109/MCOM.2011.5936160* *Sterbenz, J. et.al. (2010). Resilience and survivability in communication networks. Computer Networks. https://doi.org/10.1016/j.comnet.2010.03.005*
Obstacles, Risks, Problems	*Incumbent methods of expression and semantic models might make a transition to interoperability difficult and expensive.*	*Interoperability is a "layered" concept, like layers of an onion, from basic connectivity through to payload semantics.*	*Non-standard data, messages and documents increase and complicate the processing requirements.*	*Some pathway service providers assert a "my platform" attitude and deliberately create incompatibilities for competitive reasons.*	*There are three contexts for infrastructure relating to node interoperability:* *1. Core connections should conform with standards (analogous to a main line railway with running rights for all train operators.)* *2. Secondary connections should meet minimum standards (like limited-purpose railway sidings).* *3. Internal infrastructure may need to diverge from standards due to legacy systems or other reasons.*

Strategic Context	*Interoperability requires an appropriate legal framework:* *Examples:* *(a) Must enable competitor access with free/libre/open specifications* *(b) Must not obstruct tech compatibility through government-issued idea monopolies: a.k.a. “patents”.*	*Assume that any node can be owned and operated by a different organization.*	*For any processor to be able to send and receive information requires both:* *Payload standards* *Transport standards*	*The system must employ a “4-Corner Model” : all users choose their own provider who then interconnects. Anything less is, by definition, not “interoperable”.* *Reference: https://www.peppolbasics.info/what-is-peppol/*	*The core network infrastructure should have cooperative not-for-profit governance, even if the members and the services they operate “on” it would be profit-driven. It should be a platform”for” profit, not a profit centre itself.*
Priorities & Sequencing	*1st: Legal framework* *2nd: Business model* *3rd: Technical design and implementation* *For technical design, start with a basic system that “works”, demo it with realistic data payloads and scenarios.*	*Create a simple modular interface serving basic needs, and add capabilities step-by step.*	*Beyond the initial proof-of-concepts, proceed to design and develop the processor through an international working group.*	*Learn about and build upon the interoperability methods that already exist and extend as needed.*	*Learn about and build upon the interoperability methods that already exist and extend as needed.*
Action: "Do" **Caution: "Don’t"**	*DO: Design a way that any computable expression of legal or business requirements is also readily understandable by non-progammers, to faciliate accurate auditable digital automation* *DO: Create video tutorials to show others;* *DO NOT: Assume suitability for others. Ask them!*	*DO: Use global standards for business and transaction data.* *Examples:* *UBL and (if necessary) its extensibility rules.* *ISO 20022 for payment data.*	*DO: Design in modular form and use proven free/libre/open components* *DO: Develop people networks, not only technical networks.*	*DO: Ensure non-proprietary core requirements and standards conformance* *Examples:* *OASIS AS4 open standard interoperability protocol for payload-agnostic exchange of B2B documents https://www.codit.eu/blog/as4-for-dummies-part-i-introduction/* *IETF BEEP (Blocks Extensible Exchange Protocol) open standard framework for creating network application protocols http://beepcore.org/*	*DO: Use the power of the Internet to its full capabilities (as it was designed; as it continues to evolve)*