Open Systems Architecture and Cyber

Bryce L. Meyer

St. Louis UNIX Users Group

08 Feb 2017

Note: Just my opinions here, not representing any employer or any other group….Not liable, no warranty, etc…..just for training purposes only. Don’t sue me.

Overview

- What is an Open Systems Architecture?
- Openness is a tradeoff.
- How do cyber concerns flow into OSA implementing systems (SoSs)?
- Examples of standards as vectors
- Ecosystem tradeoffs in cyber
- (Risk) mitigation
- Reading list

What is an Open Systems Architecture?

- (IMHO :) Open Systems Architecture (OSA): A set of decisions and standards that define a structure to define a system or system of systems that:
  - Makes a system or system or system with reusable and interchangeable software/hardware components from various sources, using standard interface types
  - Creates a framework for an ecosystem of developers, manufacturers, users, testers, and maintainers.
  - Uses and enforces standards common to a large community, including standards for interfaces (software, network, hardware) and data structures
  - Allows a diversity of developers/manufacturers/groups to influence emerging framework

  OSA is not the same as Open Source! Many OSA use zero Open Source, and many are hardware/electronics focused.

  OSA is a more formal Open Architecture…?

A more official Definition: http://www.navy.mil/navydata/fact_display.asp?cid=2100&tid=450&ct=2

Who Uses OSA? EXAMPLES

- (DoD) Future Airborne Capability Environment (FACE) (https://www.opengroup.us/face/)
- (DoD) VICTORY Open Vehicle Architecture (http://victory-standards.org/)

- Non-DOD Standards that enable OSAs:
  - SAE J1939 (http:// www.sae.org/misc/pdfs/J1939.pdf)
    - Connects various vehicle processors and electronics from various manufacturers together and allows data interchange)
  - NMEA2000® Standard (using ISO 11783-3) http://www.nmea.org/content/nmea_standards/nmea_2000_ed3_10.asp
  - ISO 11898-2:2016 and TC22 Standard series (i.e. the Controller Area Network CAN on your car)

Making an Open Systems Architecture

- What problem am I trying to solve?
  - Usually I am making a family of systems or a complex system (of systems).
  - Notional Example: A class of Starships, systems.
    - Note: Many NON VEHICLE examples exist, ex: Desktop PCs, but I like vehicles…
- What components of the system (of systems) should be sourced from multiple sources?
  - Software Components?
  - Hardware Components that network or connect together?
- What data needs to be shared to make the system work?
- How big is my community of suppliers, users, maintainers?
- What commonly available standards answer these questions?

Notional Starship Navigation, Support, and Control System Open Systems Architecture

- My Starship will have many systems that plug together…
  - All components will use redundant IEEE802.3 Gigabit Ethernet to communicate with my Bridge Computer System using TCP/IP.
  - All will send data using XML (Schema defined under SomeOpenStandard that includes queries in SQL)
  - All will use 1GWatt Power Buses
  - Every Device will have its IP Addresses Assigned (using some open standard)
  - Bridge Computer Modules will also be linked using IEEE802.3/XML
  - Services and Apps on Crew Tablets using the Bridge Computer Module will use (some App/Service structure like Android or Ozone, etc.)
  - Etc. Etc. Etc. All Architecture elements published on the galactic internet…
- There is a group of members that tests all elements for conformity to the standards…

Why? OSA allows:

- competition for components
- A wide variety of new applications
- A synergy due to the ecosystem
- Economies of Scale
- Avoidance of Vendor Lock

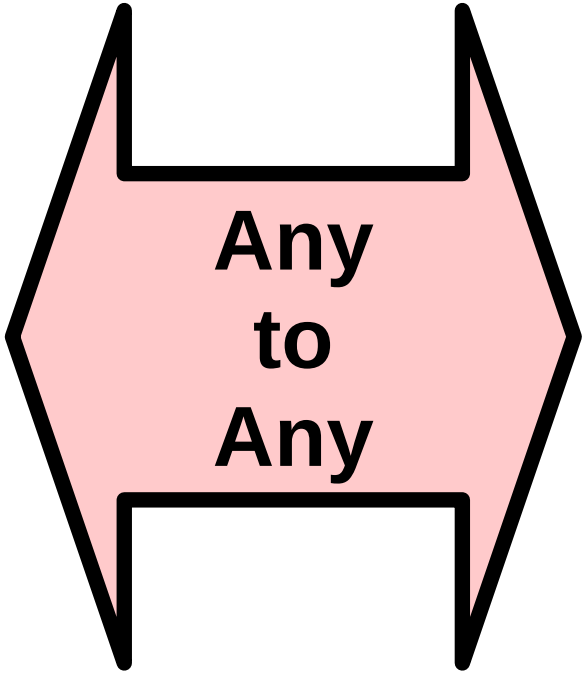
BUT…There are concerns (especially in cyber):

Adversaries: Who Wants In and Why

Actor x Motivation

Political
-Nation-State
-Terrorist Groups
-Individuals
-Loose Organizations
Fame
-Hobby Crackers
-Script Kiddies
-Professionals
Fortune
-Corporations
-Organized Crime
-Simple Criminals

Combinations of Above



Outcome Sought

Disruption
-Prevent use or delay response
-Hide or falsify information and movements
-Destroy and Defame/Deface
-Redirect to causes
-Affect Elections or Legal Case Outcomes
Intelligence
-Technological Gain
-Find Locations, Movements and Actions
- Get Identity Data
Financial Gain
-Intellectual Property
-Stock Manipulation
-Privacy or Personal Information
-Theft and Extortion (ex: Ransomware)
-Illegal Transactions and Dark Web
-Sales or Product Advantage
Fame
-Reputation
-Attention to self, group, or cause
Combinations of Above

How Are Open Systems Architectures (OSAs) Vulnerable?

Openness is a two way door. OSA:

Disseminates methodology to a community, either widely open or to a known group

Enables an ecosystem of users, implementers, observers. Ecosystems emerge from the exchange of ideas and resources due to a diversity of needs and resources. They can be fostered by communication and shared standards/needs/uses.

- The more open the architecture, standards, and software used, the larger the ecosystem

Architecture incorporates decisions and standards. Standards may be open or proprietary (and endemic to the architecture, or more global in nature).

Architecture defines separation and interfaces. Interfaces conform to architecture standards.

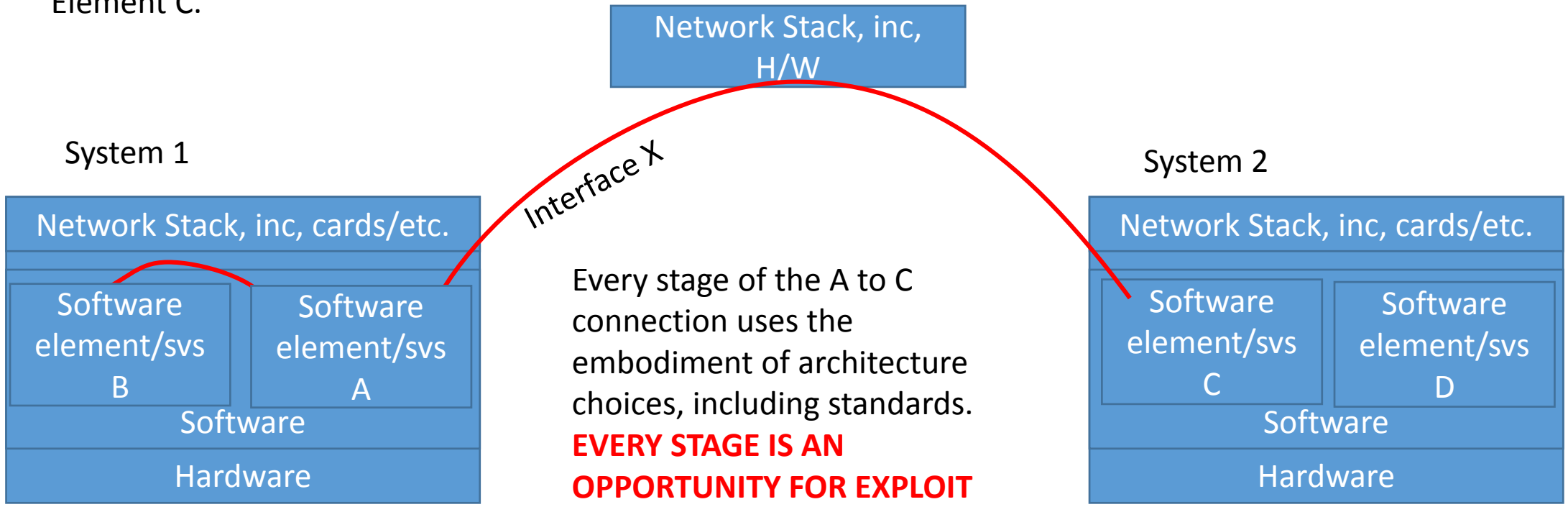
How Are OSAs Vulnerable? (Cont'd)

Each OSA implication opens opportunity for adversaries:

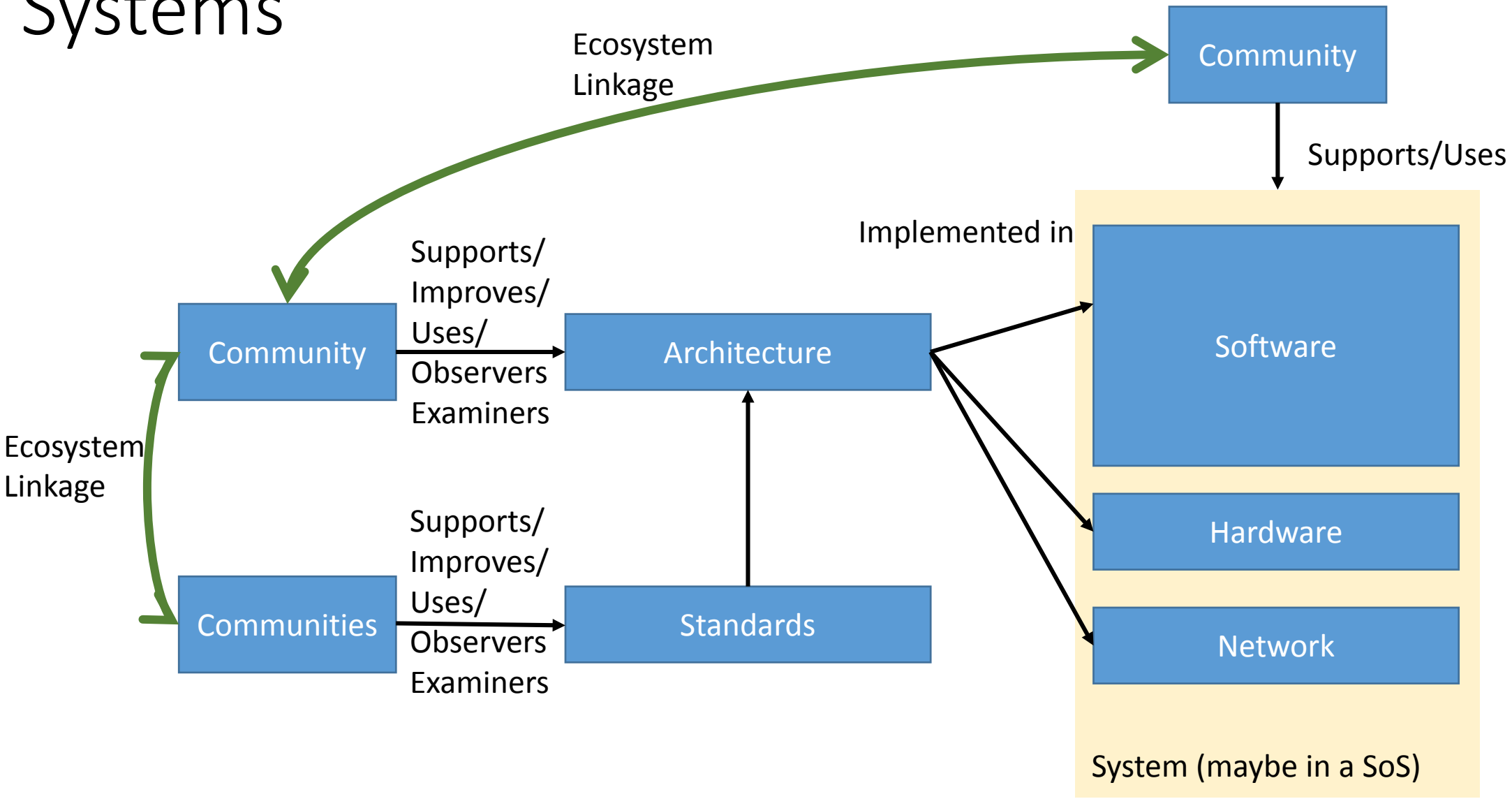
- Ties to standards provide an initial avenue of study to find vectors
- The ecosystem itself can be compromised
  - The community may implement system components under various levels of rigor or security
  - Each component may itself contain an exploit
  - While more eyes can find issues, more hands can add them (accidentally or not)
- Every interface is also a vector. It's very design is to move information into and between elements. Separation only works as well as the interfaces that use the element.

OSA Into A System of Systems: The Stack

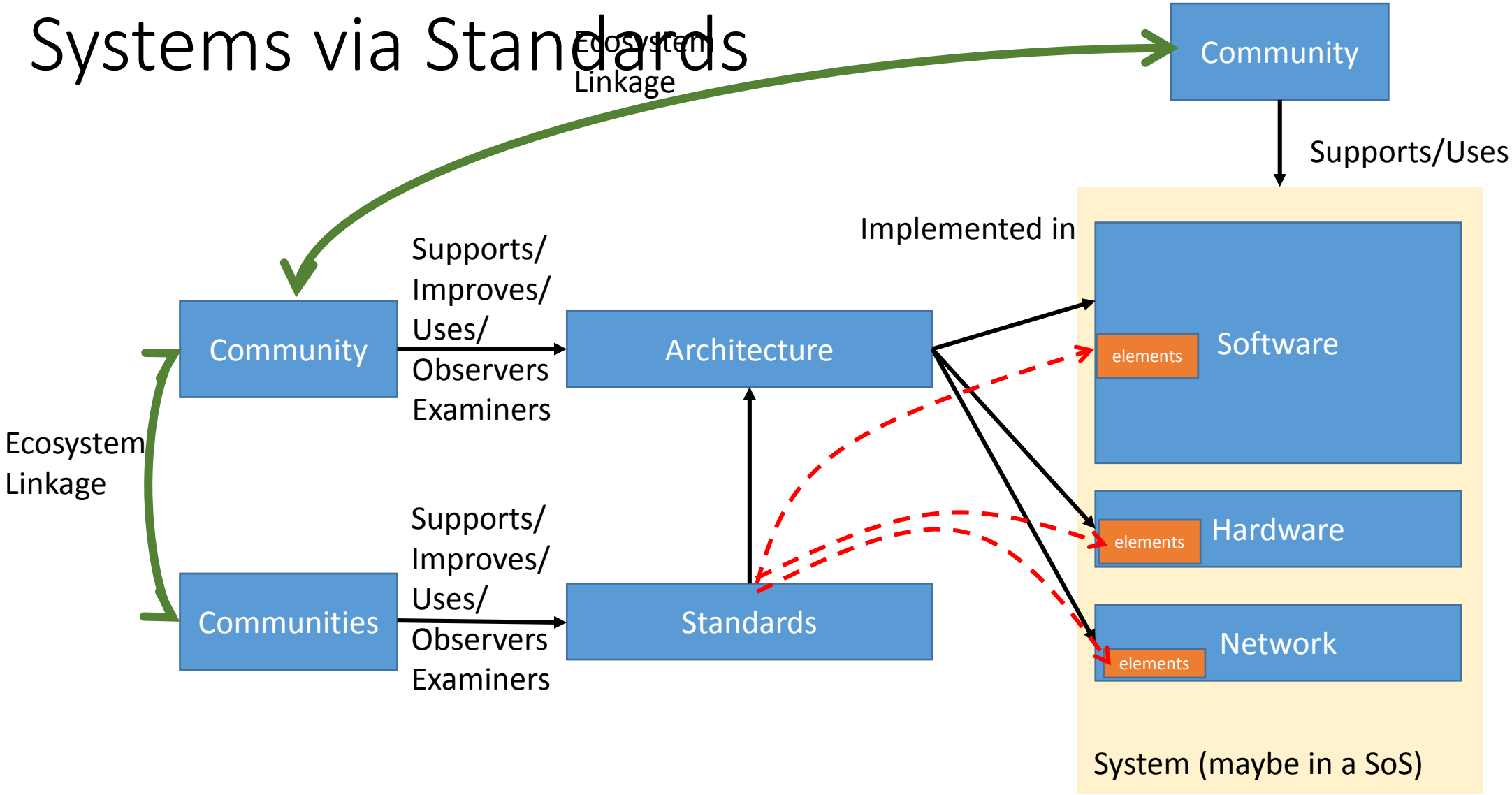
- Software Element A implements a series of standards listed as part of the architecture, including standards and architecture choices in Interface X.
- Software Element A using X to C uses and crosses operating/infrastructure software on processing hardware on System 1, uses the network interface cards and software, passes through the hardware, medium (i.e., fiber, EM Spectrum), and software of the network itself, then enters System 2's network interface card and software, through the System 2 operating software and hardware into software Element C.



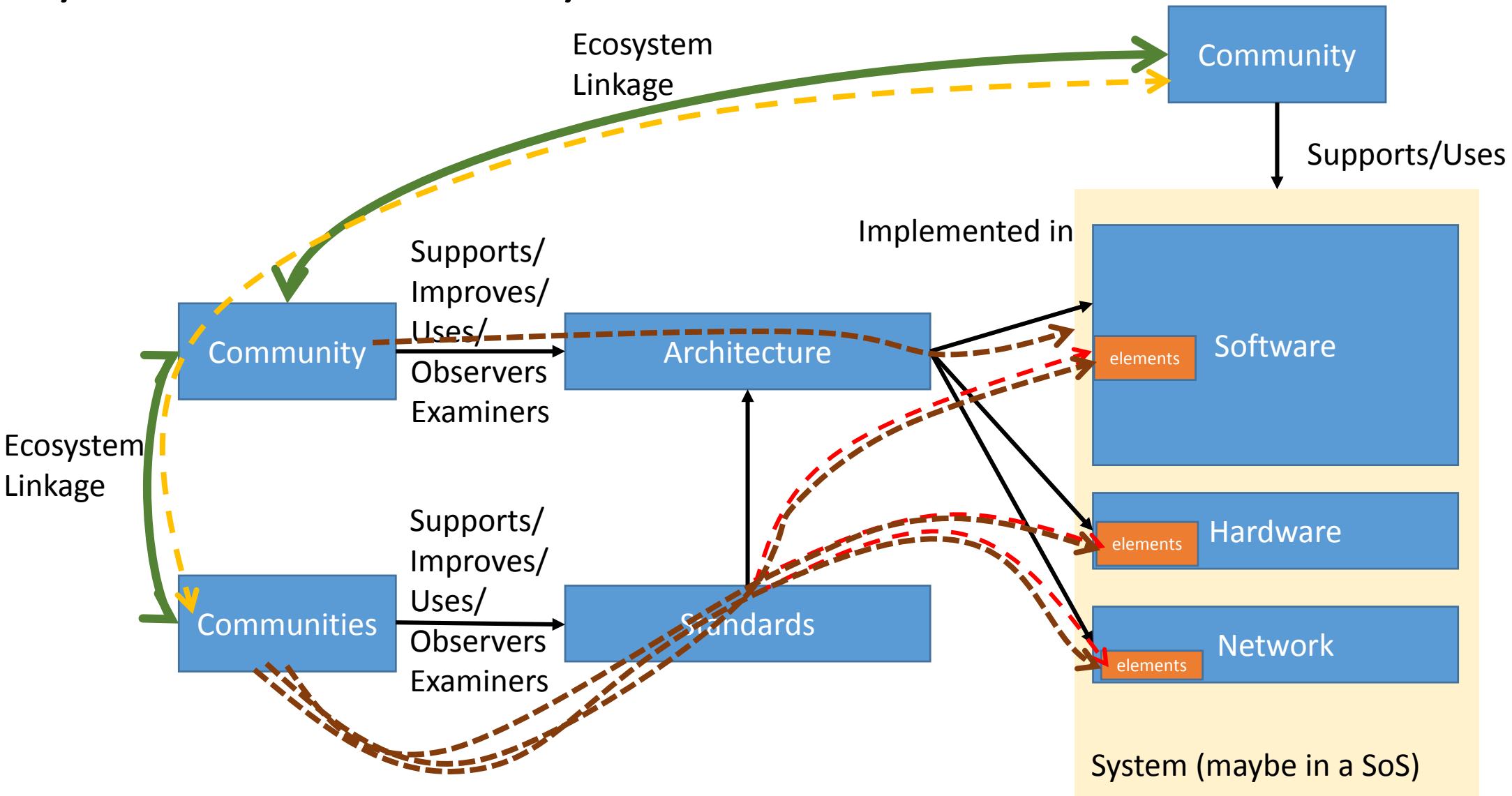
Flow From Open Architecture to Implemented Systems



Flow From Open Architecture to Implemented Systems via Standards



Flow From Open Architecture to Implemented Systems via Ecosystem



“Standard” Exploits

- USB (peripheral): Any system that implements USB can load malware which grants control of the reading device to someone else. This doesn’t include using it to introduce other exploits in the software or via connection to compromised USB chipset.
- XML: XML usually invokes schema and document definition. XML can include exploits. Schemas and definitions can be swapped. Can become pervasive in a heavily used XML for either data transfer or interface code generation.

More Standard Exploit Examples

- Smartcard Standards: Chip interacts with software and hardware to send credentials/keys in a reader. Software in reader can be compromised also. Either one allows abuse of the credentials for later use, or direct capture of information pulled using the SmartCard.
- Encryption Protocols: Encryption methods have a large variety of possible exploits. Most involve either key capture via spoof or collision. Allows anything encrypted to be partially or totally decrypted.
- Bus or Board Standards can include chips that are constructed to standard, but contain exploits. These exploits can grant control.
- PDF, GIF: Both file types can be used to import malware to any software that reads them by standard.
- SQL, Ports, Network Protocols, etc.

Experience Using Software in Standards

- Implementers of architectures using standards will likely use off-the-shelf or open source libraries or software to implement the standards, due to economic tradeoffs, or simple expediency (copy and paste at a min.). This can be dangerous.
- Commonly used packages are tempting targets for exploiters to examine and hack (especially interface implementing S/W). Obfuscation is a way to hide a hack in lines of free or xOTS code.
- “Free” or commercial software itself may collect data on users, add exploits to implementing system elements, or add vectors by accident due to poor quality.

Ecosystem Tradeoffs

- Large diverse ecosystems drive innovation. They also drive both robustness AND exploits.
  - More “eyes” can find more exploitable issues
  - More “eyes,” however, increase the risk of including bad actors
  - Tradeoffs should be managed by process and compliance. Usually the trade in favor of larger communities (IMHO) is positive.
- Law of Entropy for Cyber: The more complex the system, process, or development, the more opportunity for errors, missteps, or missed items in reviews, which in turn leads to more attack vectors.

Risk Mitigation

Understand the communities for included standards, for the architecture, and implemented system users.

Commitment to standards enforcement, and continuous maintenance of the architecture and models. Require the same on participants in the OSA.

- Leadership/governance process helps.

Review of inclusions when implementing architectures

Vulnerability monitoring group and process for standards in the OSA community

Robust quality process at all stages of implementation, and requirement for same for those groups wanting to be included in an architecture implementation.

List of vulnerable or bad implementations and practices for the architecture.

Conclusion

- Open Systems Architecture (OSA) is a structure that marries standards and design to allow interchangeability and foster large ecosystems to benefit system (of systems) makers.
- Openness has benefits and risks. Know your architecture, its standards, and likely implementations.
  - Community size affects the benefits and costs of standard use.
  - Interfaces work to not just send and receive data, but risks as well.
- Ecosystems of communities can contain risks, mitigated by compliance and checking.
  - Compliance against selected standards is a best practice for not only security, but overall quality.
  - Governance and communication foster interchange and enforce checking.
- A side effect of risk mitigation can be improved overall quality.

Reading List:

- Open Systems Architecture: Progress and Challenges http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=447404
- Open System Architectures: When and Where to be Closed https://insights.sei.cmu.edu/sei_blog/2015/10/open-system-architecture-when-and-where-to-be-closed.html
- A Discussion on Open-Systems Architecture https://insights.sei.cmu.edu/sei_blog/2015/11/a-discussion-on-open-systems-architecture.html
- The Importance of Automated Testing in Open Systems Architecture Initiatives https://insights.sei.cmu.edu/sei_blog/2014/03/the-importance-of-automated-testing-in-open-systems-architecture-initiatives.html
- Cyber Threat Modeling: An Evaluation of Three Methods https://insights.sei.cmu.edu/sei_blog/2016/11/cyber-threat-modeling-an-evaluation-of-three-methods.html
- OSA: 4 Best Practices for Open Software Ecosystems https://insights.sei.cmu.edu/sei_blog/2015/11/osa-4-best-practices-for-open-software-ecosystems.html
- Navy Fact file: Open Systems Architecture: http://www.navy.mil/navydata/fact_display.asp?cid=2100&tid=450&ct=2
- Open Architecture in Electronics Systems: http://www.dtic.mil/ndia/2008systems/7453Bardell.pdf

Where to Find Threats and Responses (examples are from here)

- U.S.-CERT https://www.cert.org
- U.S. DHS: https://www.dhs.gov/topic/cybersecurity
- U.S. DOJ: https://www.justice.gov/usao/priority-areas/cyber-crime
- The sites for all the standards and software/hardware you are using….