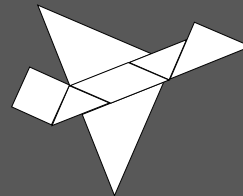
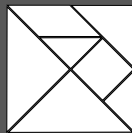


TANGRAMFLEX 

Component-Oriented Engineering

John Launchbury
CTO, TangramFlex
Chief Scientist, Galois

Building the Hub for Re-engineering Cyber-Physical Systems



More than 70% of systems are software
F-35 has more than 8 million lines of software code
DoD spends ~\$36B on software R&D each year

CHALLENGE

Rapid acquisition organizations need to re-use existing systems to support evolving missions with reduced development and acquisition time

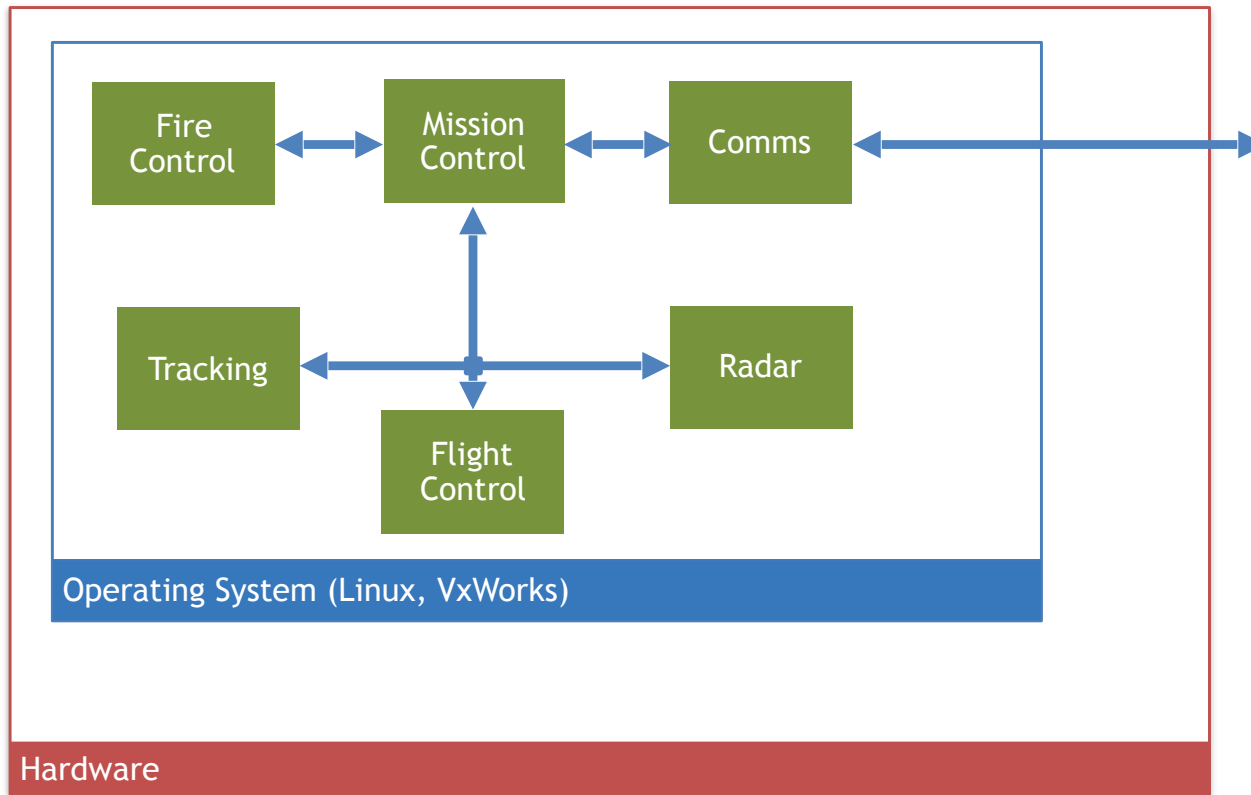
70% of errors are introduced in design
Less than 20% of errors are found prior to system testing
Bugs found late in testing cost 80-1000x more to repair

Cyber vulnerability...

Perpetrating a cascading breach

Cyber attackers

- Exploit external component to gain an initial foothold
- Find ways to pivot from one system to another
- Repeat until they gain access to critical controls

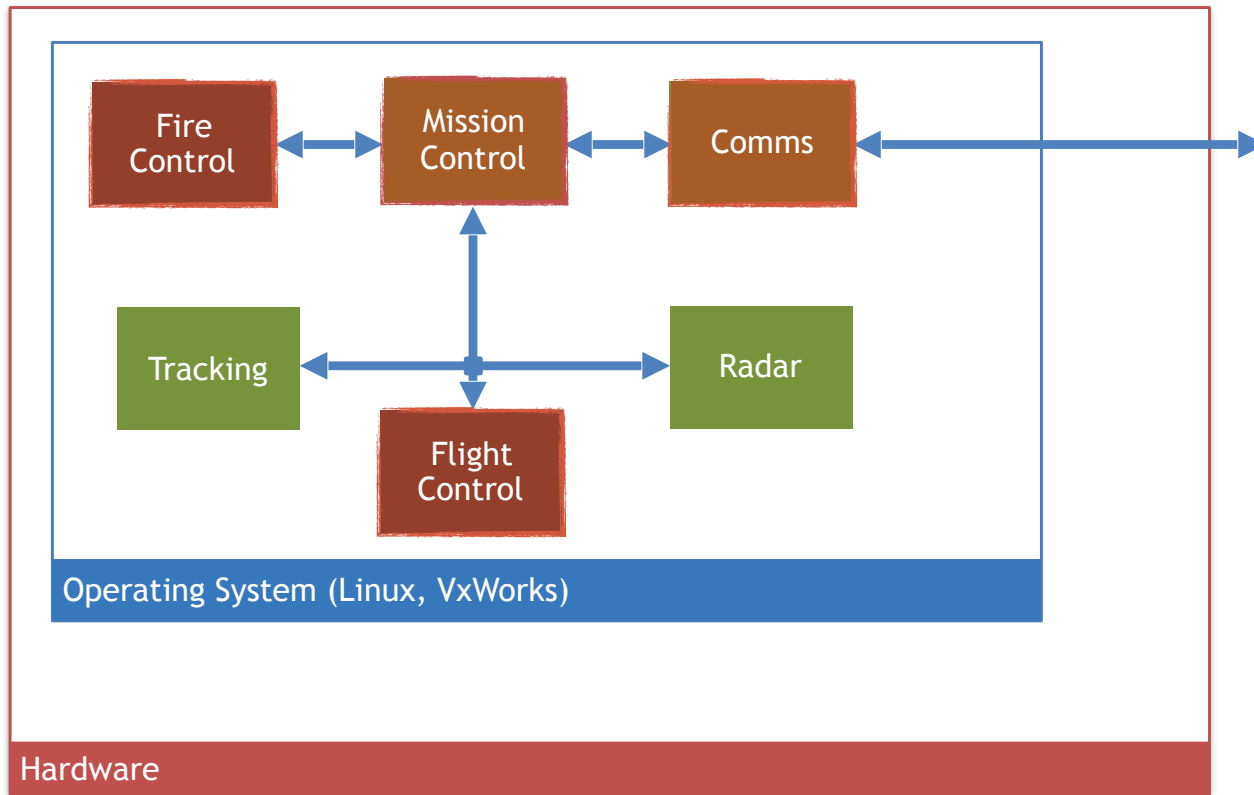


Cyber vulnerability...

Perpetrating a cascading breach

Cyber attackers

- Exploit external component to gain an initial foothold
- Find ways to pivot from one system to another
- Repeat until they gain access to critical controls



Cyber vulnerability...

Preventing a cascading breach

Limit exposure to input

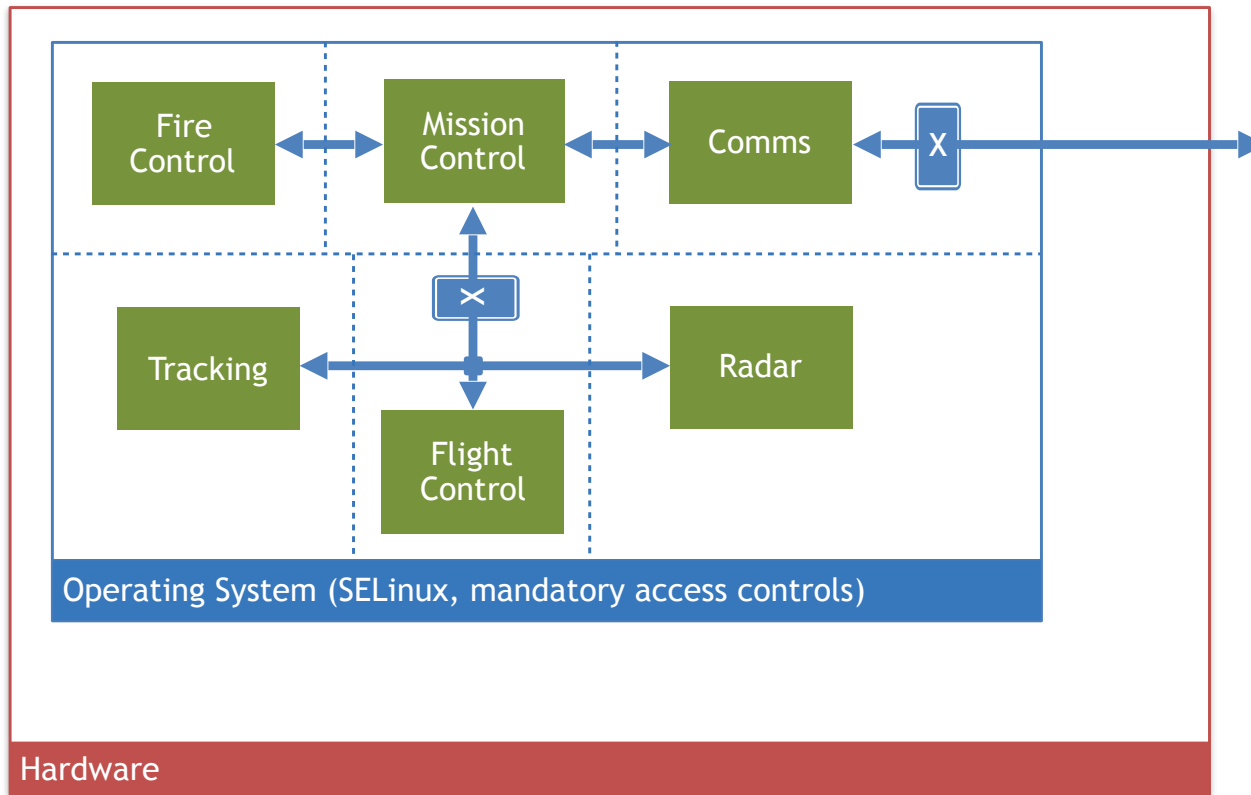
- Add filters and crypto between networked systems

Limit opportunities to pivot

- Improve separation between components

Improve code resilience

- Replace critical components with secure alternatives



Cyber vulnerability...

Preventing a cascading breach

Limit exposure to input

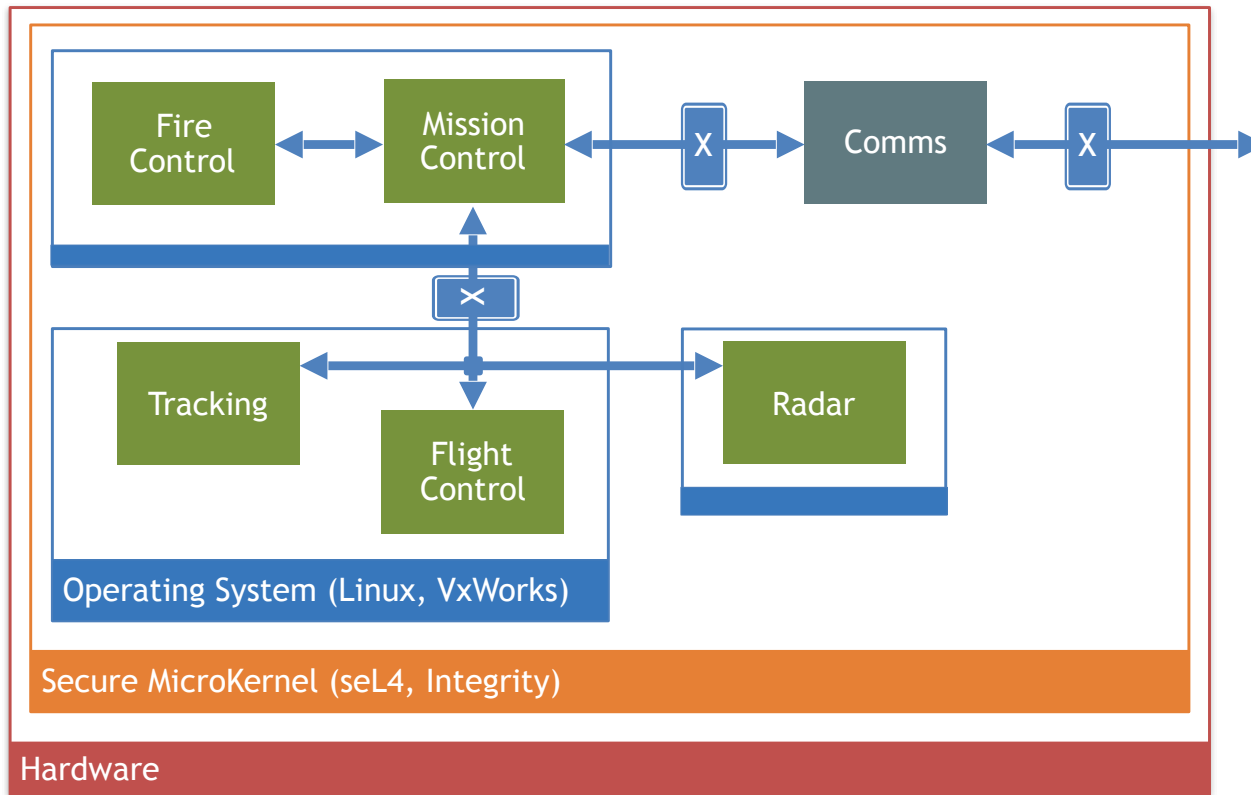
- Add filters and crypto between networked systems

Limit opportunities to pivot

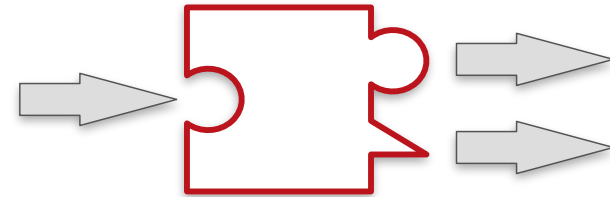
- Improve separation between components

Improve code resilience

- Replace critical components with secure alternatives



Component-Oriented Engineering...



What is a COMPONENT ?

- A stable unit of functionality that communicates with streams of messages
- May have a physical aspect, always contains software
- May be a subcomponent of a larger composite component
- Transport mechanism of messages is separable from the component







Components have a precise conceptual inside/outside boundary

Components can have clearly defined interfaces

Components are a natural unit of reuse!

DoD has been driving Open Systems for many years (OMS, FACE, UCI, etc.)...

Joys and Agonies of Open-Systems

-  Manufacturers no longer assert proprietary claims over their interfaces
-  Component-reuse becomes possible in principle
-  Standards are maintained by communities of interest and updated to reflect needs
-  Every manufacturer has their own interpretation of the standards
-  Extensive “accidental” complexity gets in the way of interoperability
-  Multiple standards co-exist and are not always fully compatible

Currently, the focus of Open Systems is on interface interoperability...

A Naval Retrospective

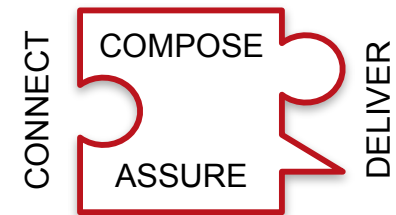
“I am starting to wonder if **the notion of open interface is too narrow**. In particular, I now believe that openness is more complicated than that. For example, what does it take to integrate a new component into a system? Perhaps it is more than just interface specifications... I have come to recognize that successful integration requires **specification of key quality attributes**, such as timeliness, security, and reliability, as well as access to technical data and related information needed to reuse components in systems beyond their original context. Ultimately, what we need is the ability to have **multiple competitive alternatives** that can evolve and interoperate dependably across the lifecycle of complex systems and systems-of-systems.”

- N. H. Guertin, Office of the Deputy Assistant Secretary of the Navy for RDT&E
https://insights.sei.cmu.edu/sei_blog/2016/07/a-naval-perspective-on-open-systems-architecture.html

Tangram provides a Software as a Service (SaaS) hub

Component-Oriented Workflow Platform

- **Connect** to existing systems and development workflows
- **Compose** system components in new ways to create new capabilities
- **Assure** that the new composed systems are safe and secure
- **Deliver** systems and certification evidence for mission use



Everything can be versioned, coupled or decoupled, independently assured, re-engineered or interchanged

To re-use embedded software, **component-oriented engineering** must solve the...

COMPOSE Challenge

Enable the seamless translation of protocols between critical abstraction layers

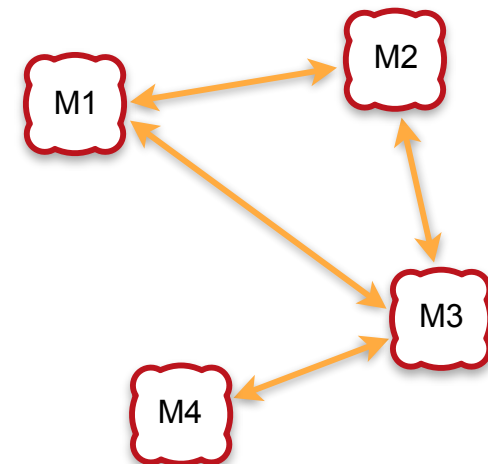
- “How do I translate my FACE (Army/Navy) component to an OMS (Air Force) component?”
- “How do I ensure consistency between DoD platforms using the same message standards...
... but using them differently?”
- “How do I provide a mechanism for industry to update Open Systems Architecture standards and critical abstraction layers without lagging behind?”

Use **automated software generation** to provide...

COMPOSE Capabilities

- Generate a Critical Abstraction Layer for a set of components
- Integrate across distinct CALS matching an open standard
- Generate a CAL that spans a set of standards
- Integrate non-standard components into a standard (e.g. packed interface format)
- Generate high assurance micro-guards to protect interfaces
- Exchange middleware transport layers

Graph of formal relationships between message formats

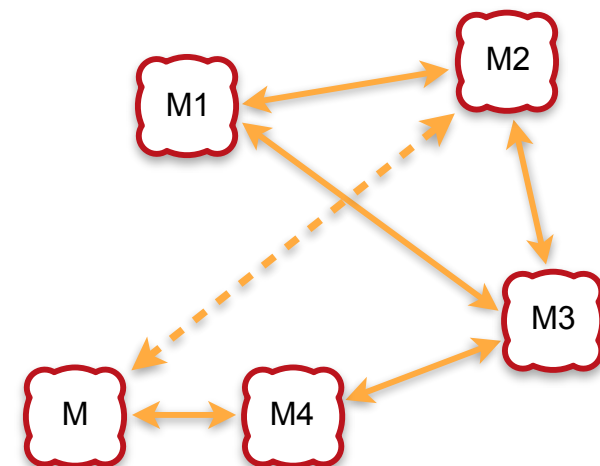


Use **automated software generation** to provide...

COMPOSE Capabilities

- Generate a Critical Abstraction Layer for a set of components
- Integrate across distinct CALS matching an open standard
- Generate a CAL that spans a set of standards
- Integrate non-standard components into a standard (e.g. packed interface format)
- Generate high assurance micro-guards to protect interfaces
- Exchange middleware transport layers

Graph of formal relationships between message formats



To re-use embedded software, **component-oriented engineering** must solve the...

ASSURE Challenge

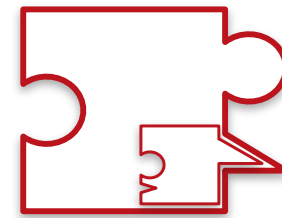
Correctness, safety, and cyber resiliency require expanding our methods of evidence generation and reusable arguments

- “How do we ‘bake in’ cyber resiliency, eliminating whole classes of vulnerabilities?”
- “How can we analyze a piece of code to understand its behavior, decompose its functionality, and prove its correctness?”
- “How do we build-in Assurance Patterns or Profiles to reduce testing burden?”

To go **beyond just interfaces** we need...

ASSURE Capabilities

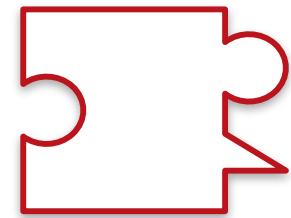
- Assist a developer to isolate a subcomponent within an existing system
- Automatically instrument its interfaces in use
- Derive interface properties for a component
- Generate evidence that a component satisfies its properties
- Build and import behavioral models of components
- Build and import models of environments
- Generate evidence that a component matches a model



To go **beyond just interfaces** we need...

ASSURE Capabilities

- Assist a developer to isolate a subcomponent within an existing system
- Automatically instrument its interfaces in use
- Derive interface properties for a component
- Generate evidence that a component satisfies its properties
- Build and import behavioral models of components
- Build and import models of environments
- Generate evidence that a component matches a model



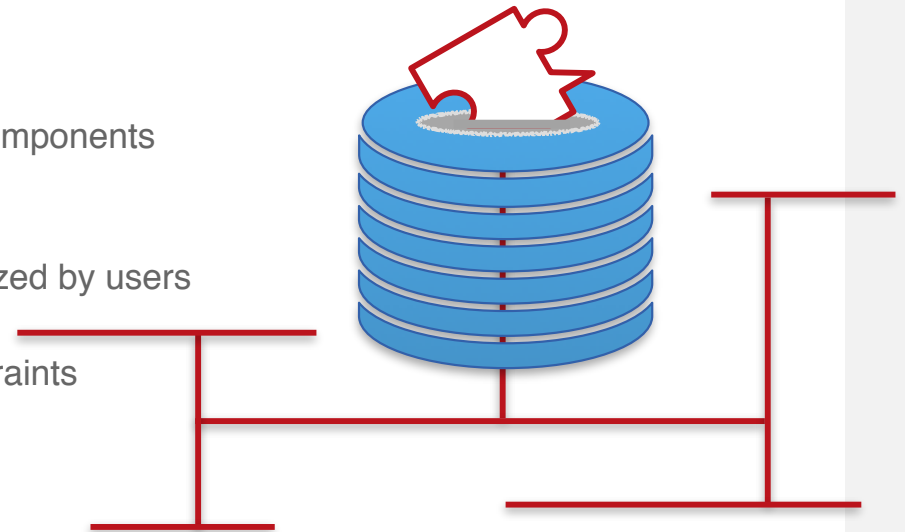
To re-use embedded software, **component-oriented engineering** must solve the...

CONNECT Challenge

Remove friction that would prevent access to reusable components

- Integrate with standard development environments
- Curate a Knowledge Base of components characterized by users
- Benefit from expanding knowledge of components
- Manage classification and intellectual property constraints

Leverage the **network effect** at scale



To re-use embedded software, **component-oriented engineering** must solve the...

DELIVER Challenge

Connect multiple build processes to deliver artifacts at each level of systems engineering

- “How do we deliver architectural standards to developers in a way that is traceable and verifiable?”
- “How do we bring a DevOps perspective to the maintenance and upgrade of components?”
- “How do we generate certification evidence for components, systems, and systems of systems?”

Component-Oriented Engineering

Cyber Retrofit

“Current systems are riddled with cyber vulnerabilities”

Radically transformed cyber security posture while reusing existing code

System of Systems

“DoD systems take 10-30 years to procure and build”

Rapid mission capabilities from new combinations of existing systems

Supply Chain

“Systems become dependent on sub-system suppliers”

Sub-system components can be swapped and upgraded with confidence

... reasons for using the Tangram SaaS platform

Sounds cool! How do I..

... use it, contribute, get involved?

- Alpha/Beta releases early 2019 – talk to us to try it out!
- Quarterly releases with feature upgrades
- Contribute components for inclusion in the knowledge base
- Contribute tools (for analysis, for composition)
- Contribute ideas/perspectives

www.tangramflex.com

TANGRAMFLEX 

Building the Hub for Re-engineering Cyber-Physical Systems