Systems Engineering Requirements Analysis and Trade-off for Trusted Systems and Networks Tutorial

Notional Architecture Handout

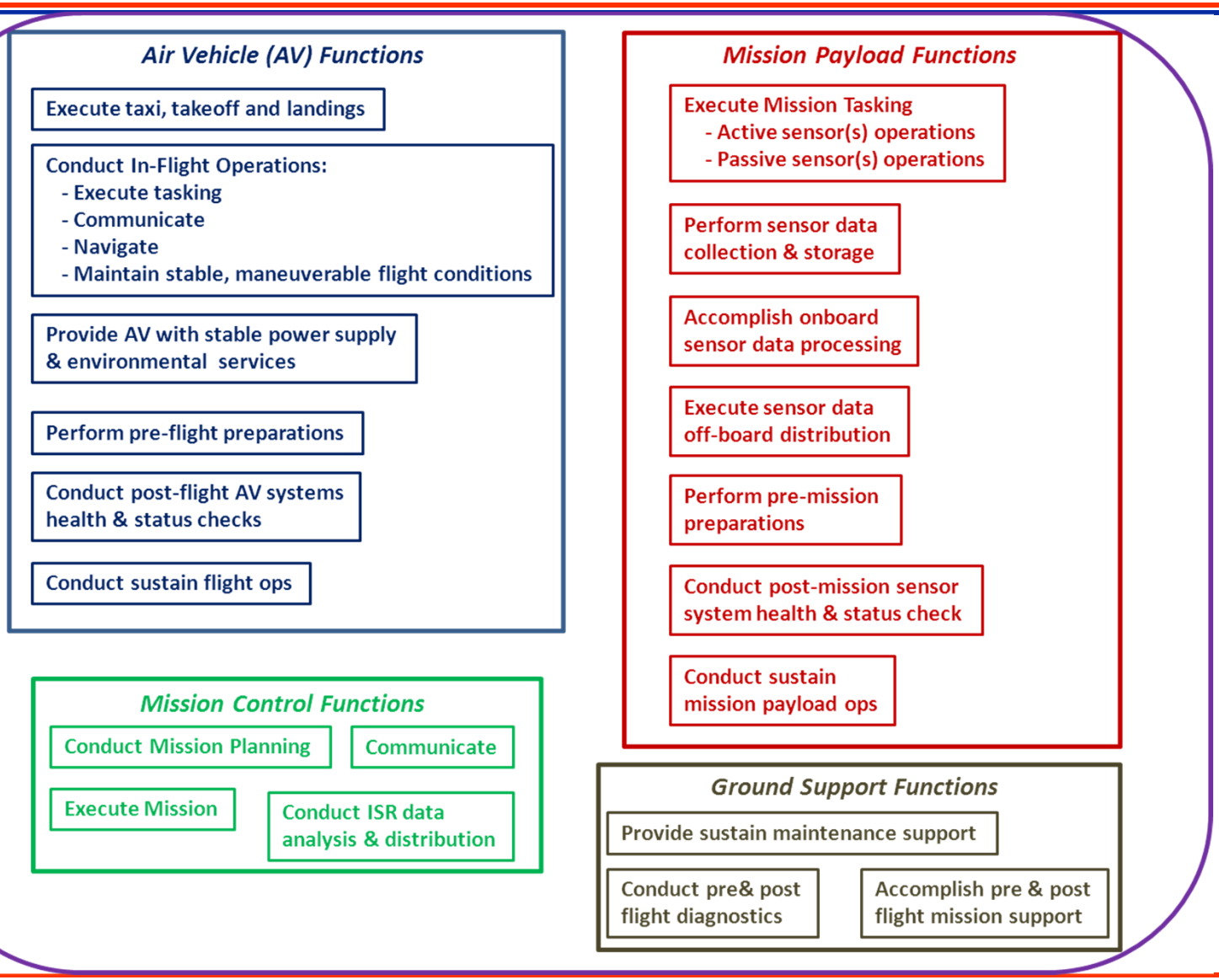
Melinda Reed
Office of the Deputy Assistant Secretary of Defense for Systems Engineering

Paul Popick
Johns Hopkins University Applied Physics Lab

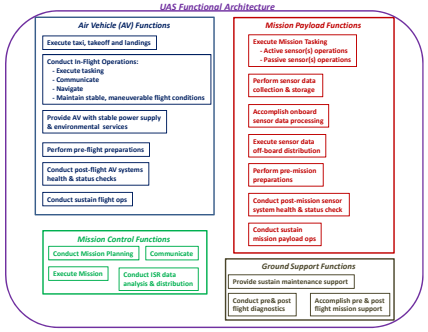
Contents

- UAS Notional Architecture
- UAS Potential Supply Chains
- UAS Potential Development Lifecycles
- Generic Supply Chain & Malicious Insertion Threats/Vectors

UAS Functional Architecture



Find-Fix-Track Scenario



Find-Fix Track Functional Architecture

- **Execute Mission Tasking**
  - **Active sensor(s) operations**
  - **Passive sensor(s) operations**
- **Perform sensor data collection & storage**
- **Accomplish onboard sensor data processing**
- **Execute sensor data off-board distribution**

Execution mission tasking functions
- Accept Mission Plan
- Allocate mission plan to sensors
- Initiate active sensor search plan ***(Search)***
- Position sensor to identify contact with passive sensor(s)
- Accept tasking to either: 1) initiate tracking or 2) return to search plan ***(Track)***
- Mission Control tasks return to mission plan execution – *next cycle in FFT kill chain execution*

Perform sensor data collection & storage function
- Collect sensor return data

Accomplish onboard sensor data processing function
- Process sensor return data
- Determine if contact is possible target or not ***(Detect)***
- Locate contact and establish position, course and speed ***(Locate)***
- Analyze for contact classification ***(Classify)***

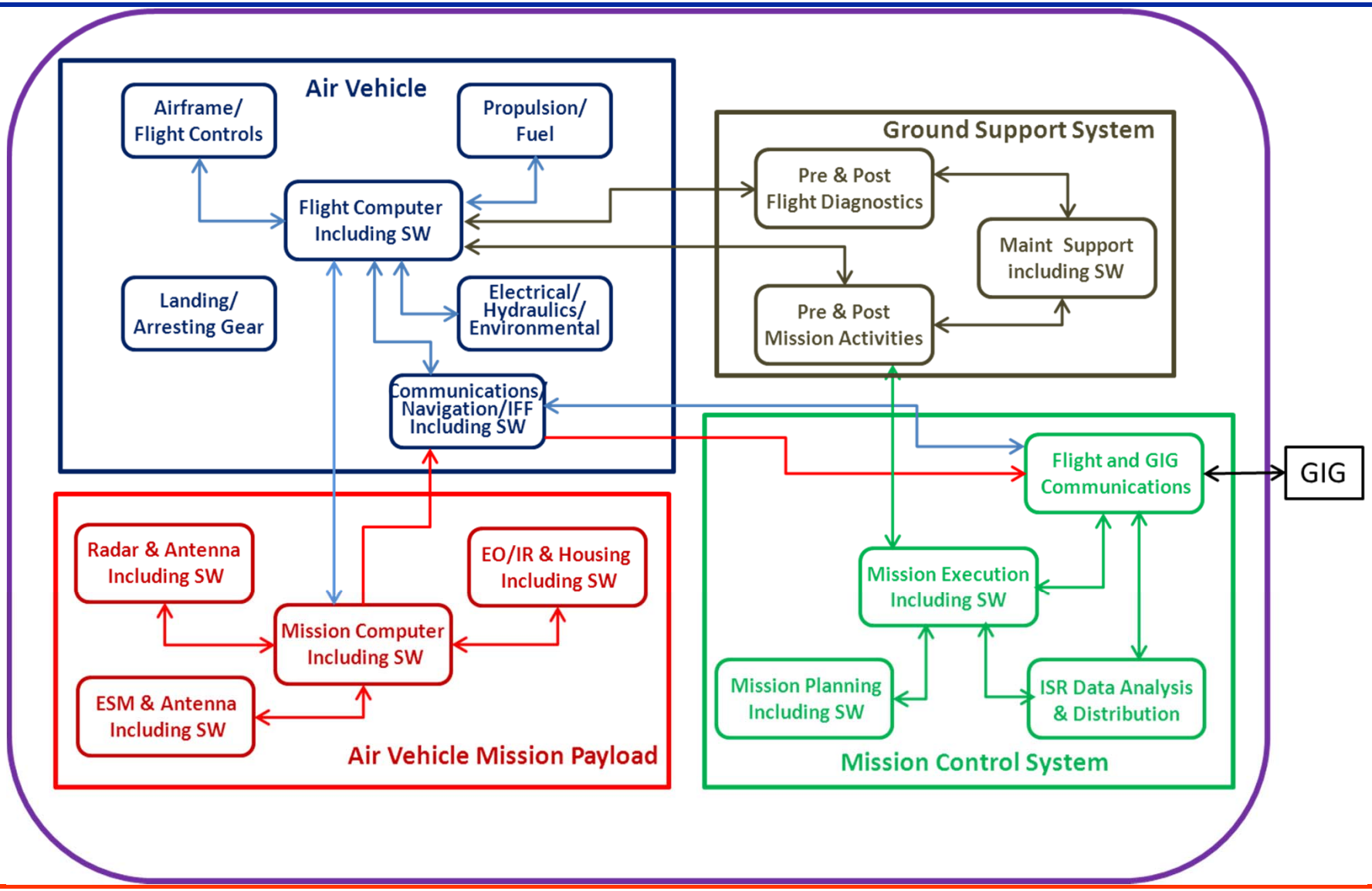
Execute sensor data off-board distribution function
- Pass sensor data and analysis results to mission control for confirmation ***(Communicate)***

Find, Fix, and Track Functional Order :

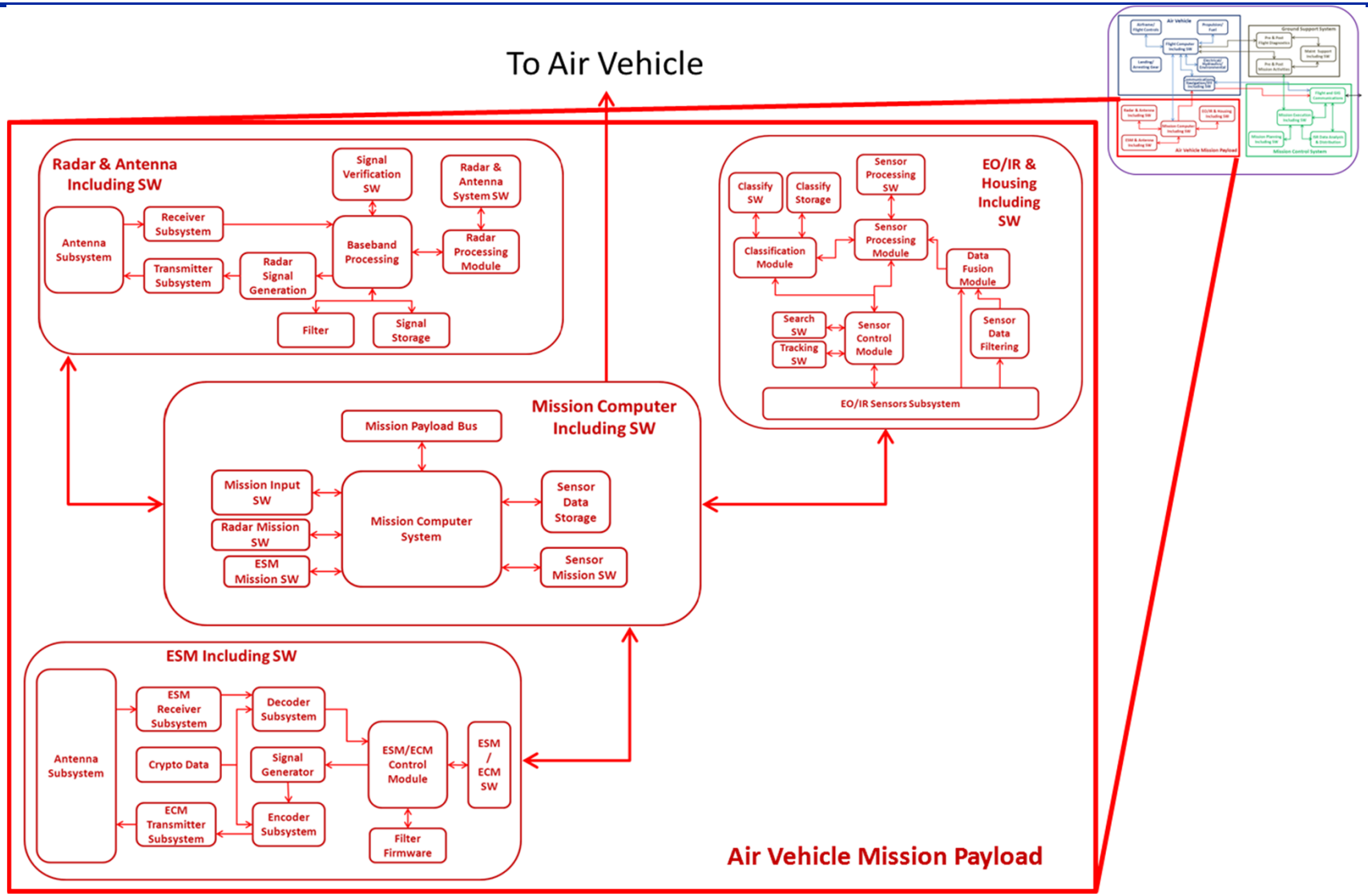
1. Accept Mission Plan
2. Allocate mission plans to sensors
3. Initiate active sensor search plan (***Search)***
4. Collect and process sensor returns
5. Determine if contact is possible target or not ***(Detect)***
6. Locate contact and establish location, course and speed ***(Locate)***
7. Position sensor to identify contact with passive sensor(s)
8. Gain passive sensor(s) data and analyze for contact classification ***(Classify)***
9. Pass sensor data and analysis results to mission control for confirmation ***(Communicate)***
10. Accept tasking to either: 1) initiate track or 2) return to search plan ***(Track)***
11. Mission Control tasks return to mission plan execution

Note: Search, Detect, Locate, Classify, Communicate and Track are mission thread functions.

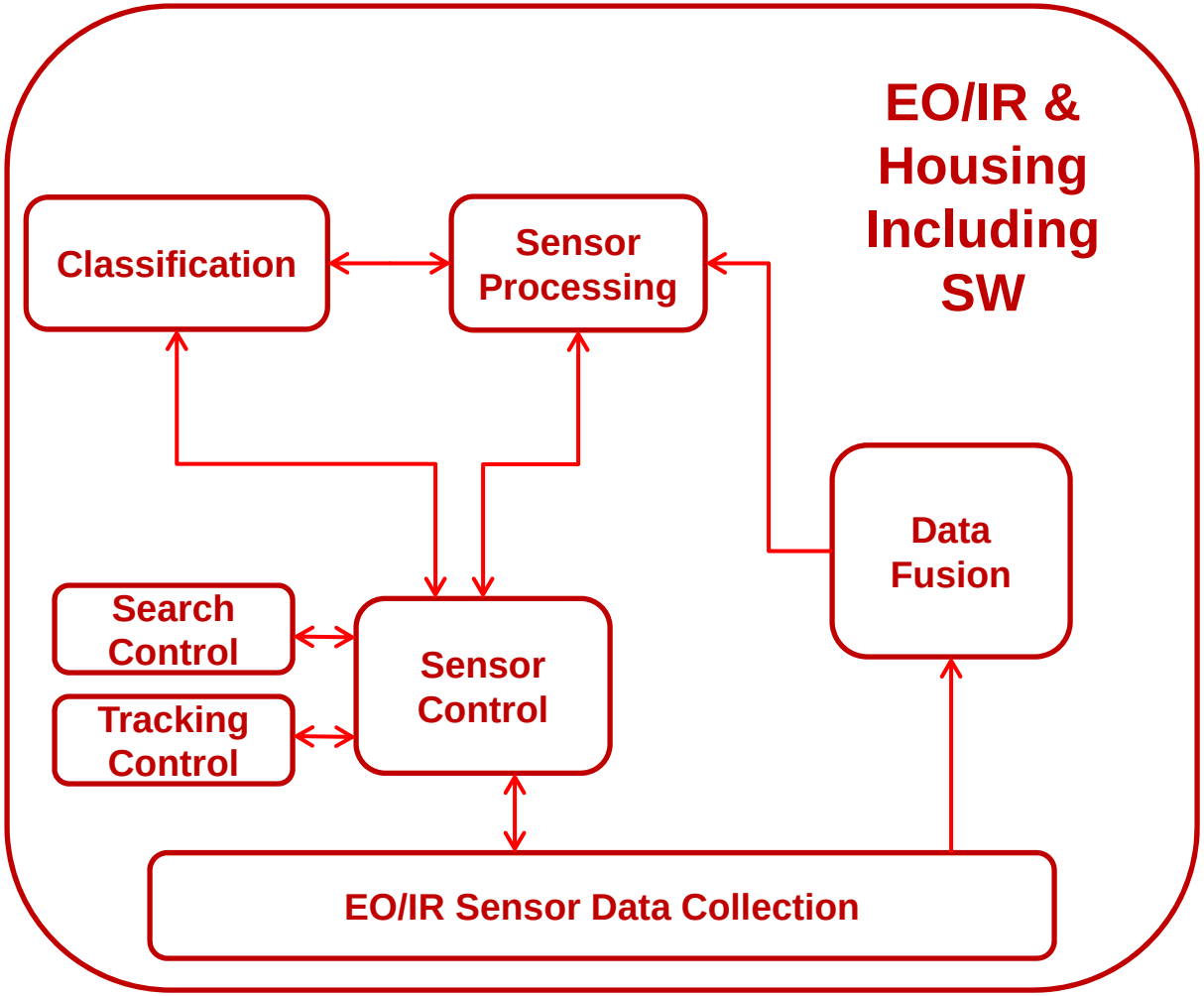
UAS High-Level System Design



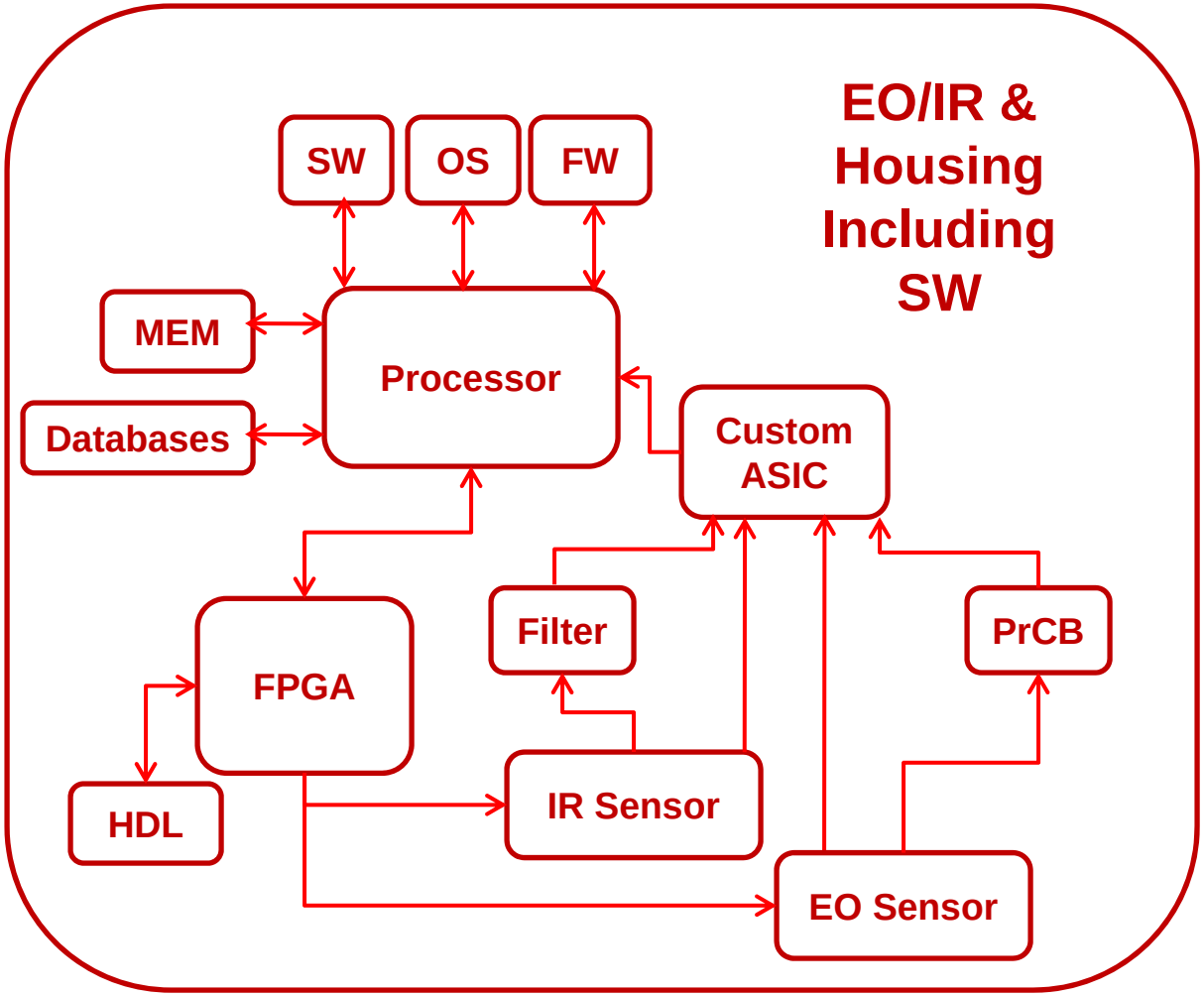
Air Vehicle Mission Payload Diagram



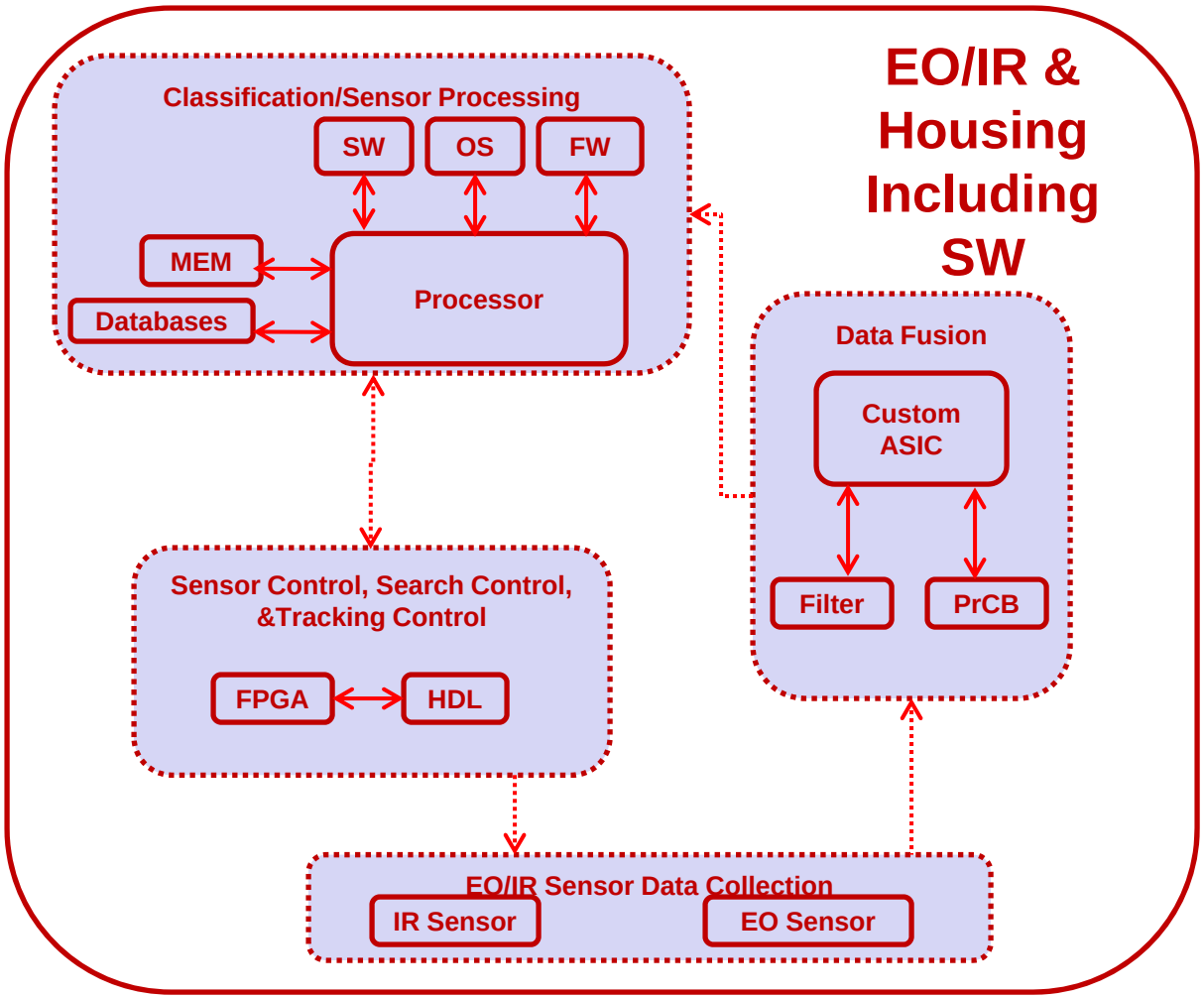
EO/IR & Housing – Functional



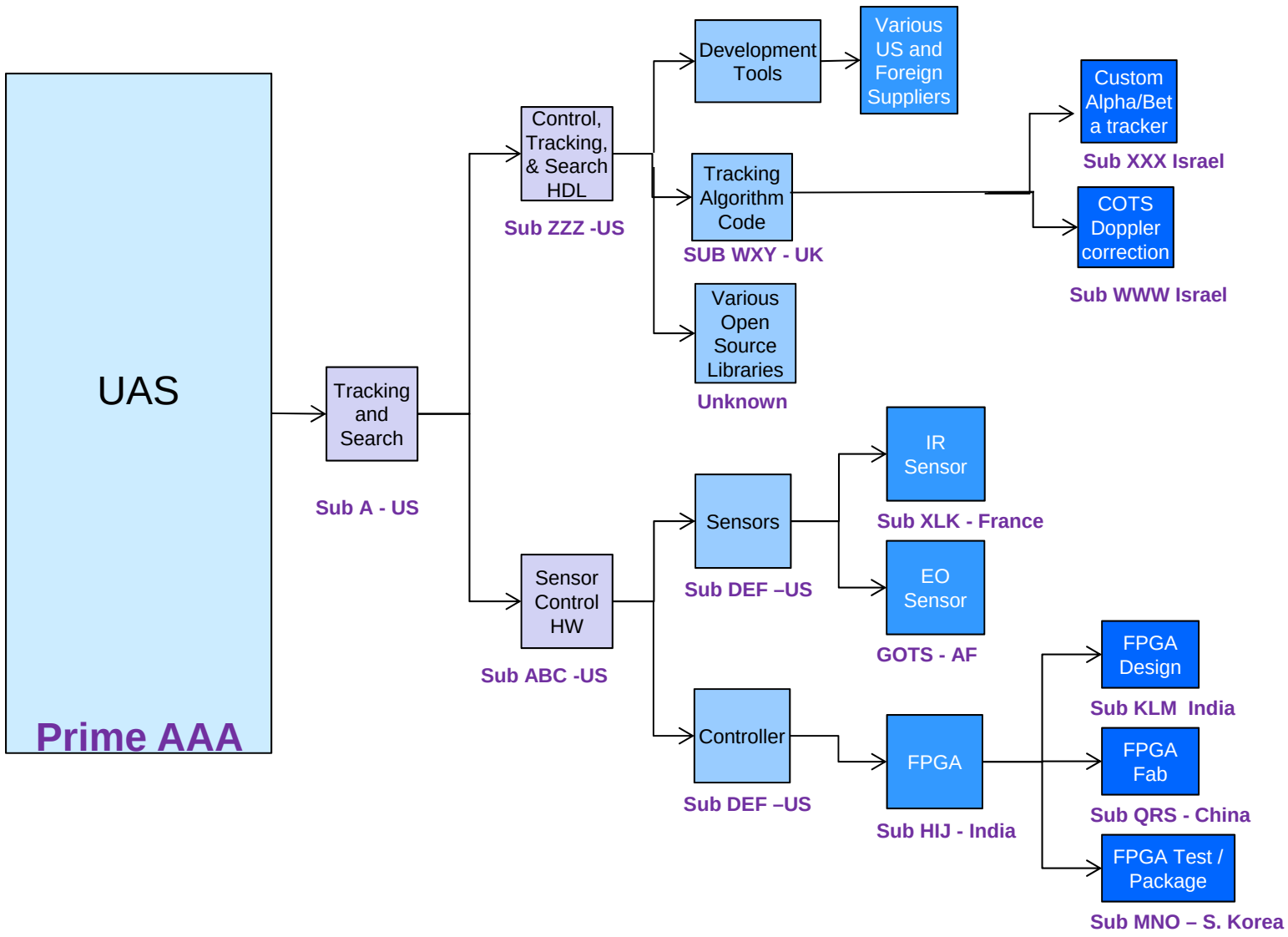
EO/IR & Housing – Physical (Supply Chain 1)



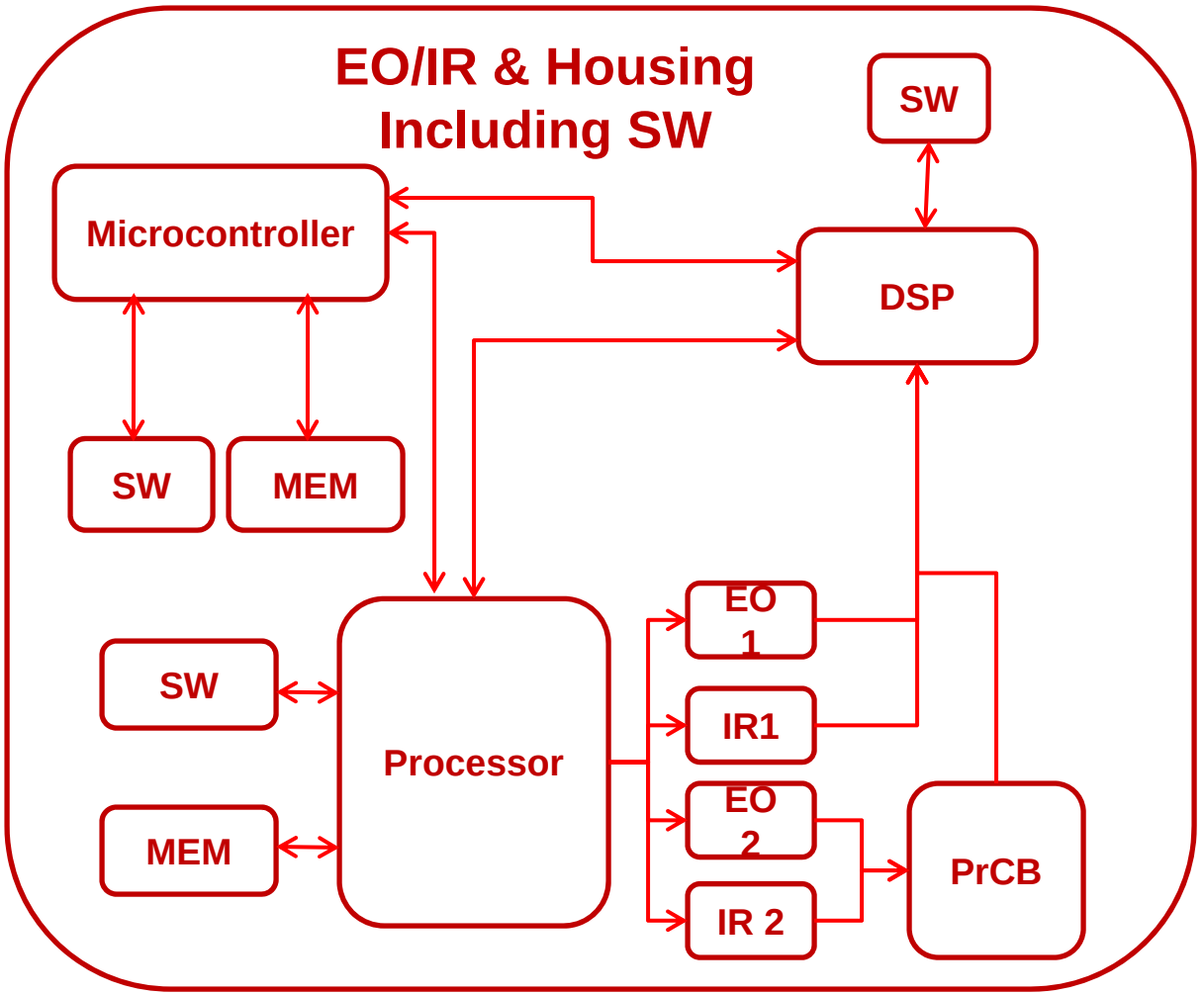
EO/IR & Housing – Allocated (Supply Chain 1)



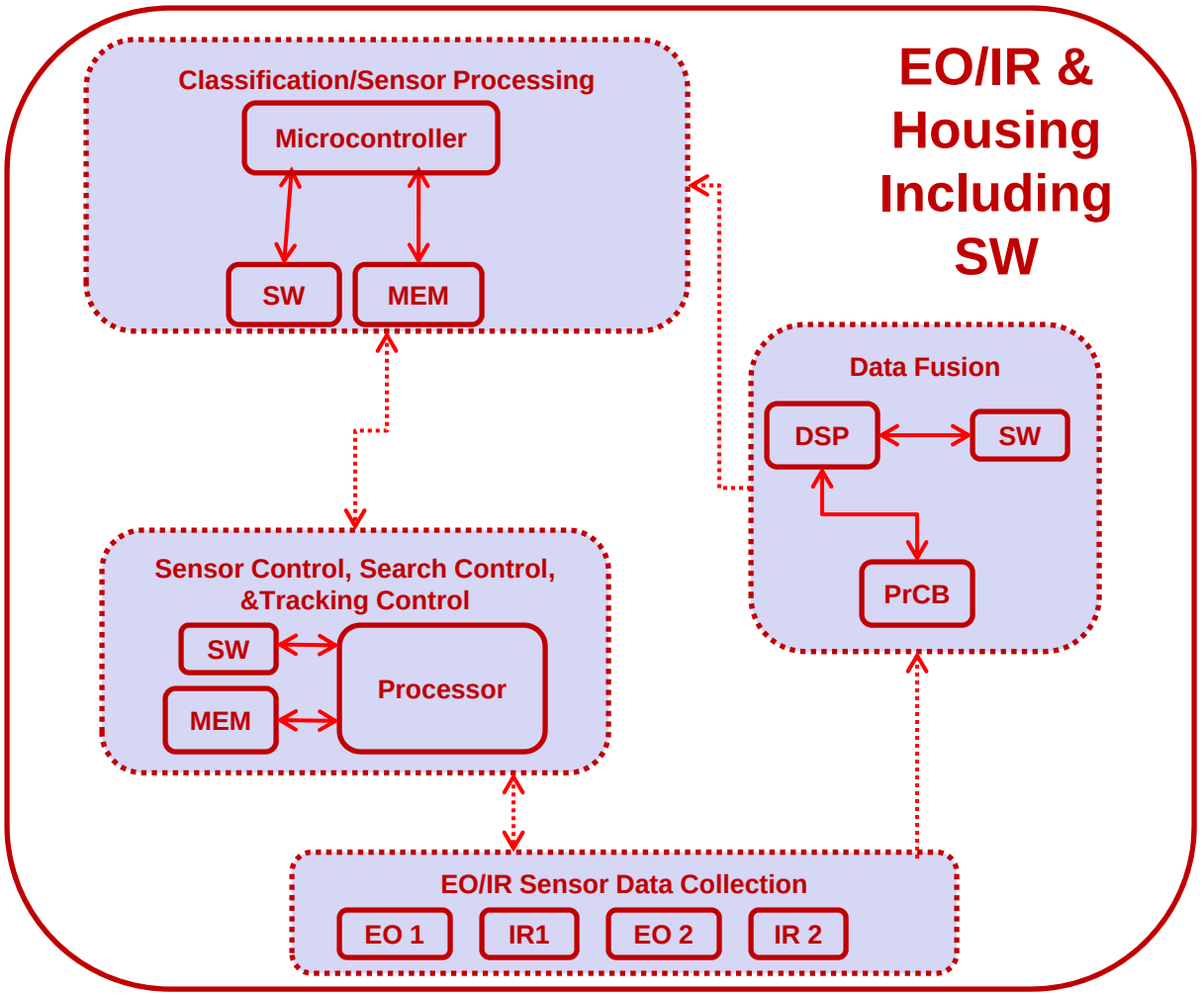
Potential Supply Chain 1



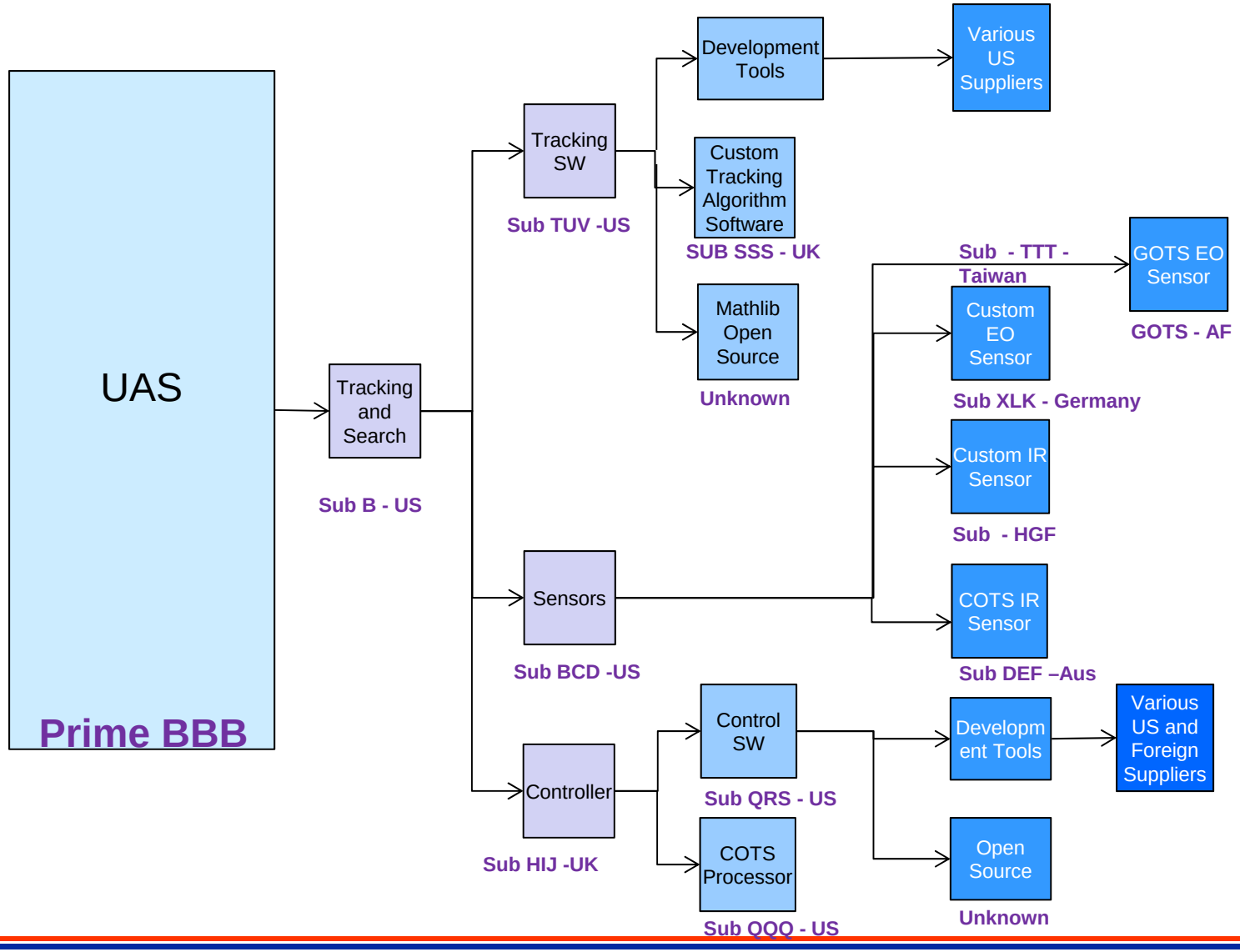
EO/IR & Housing – Physical (Supply Chain 2)



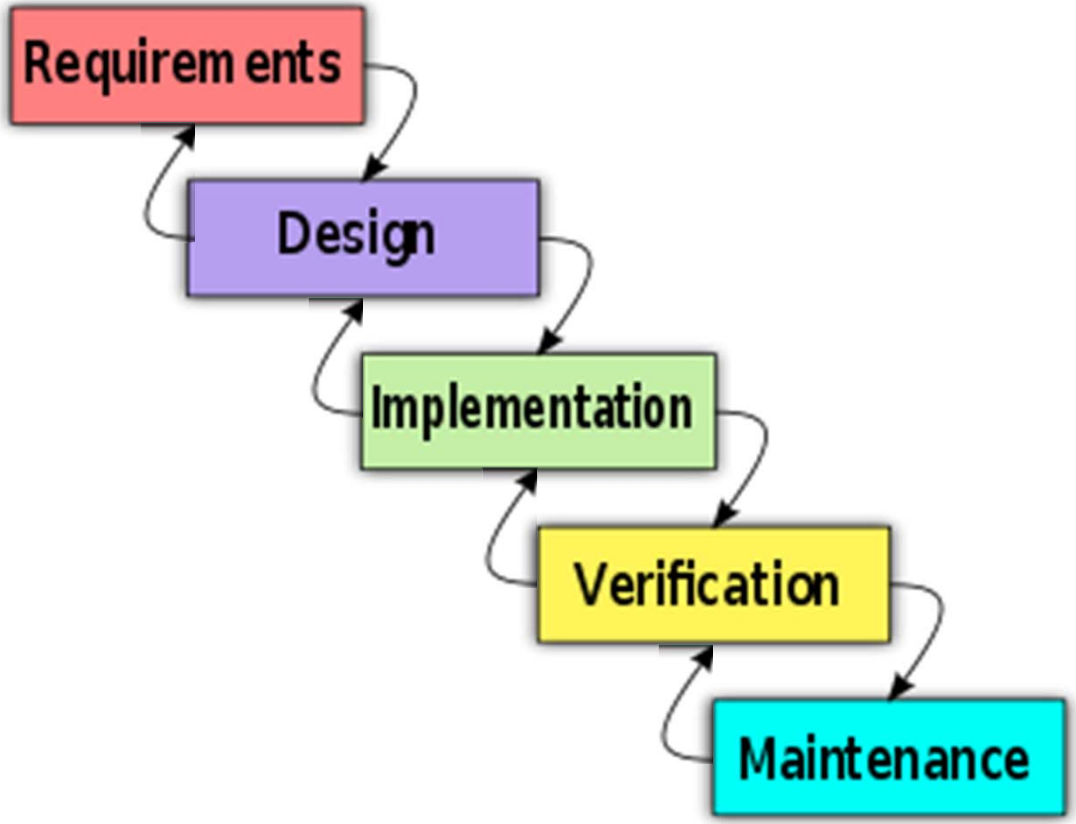
EO/IR & Housing – Allocated (Supply Chain 2)



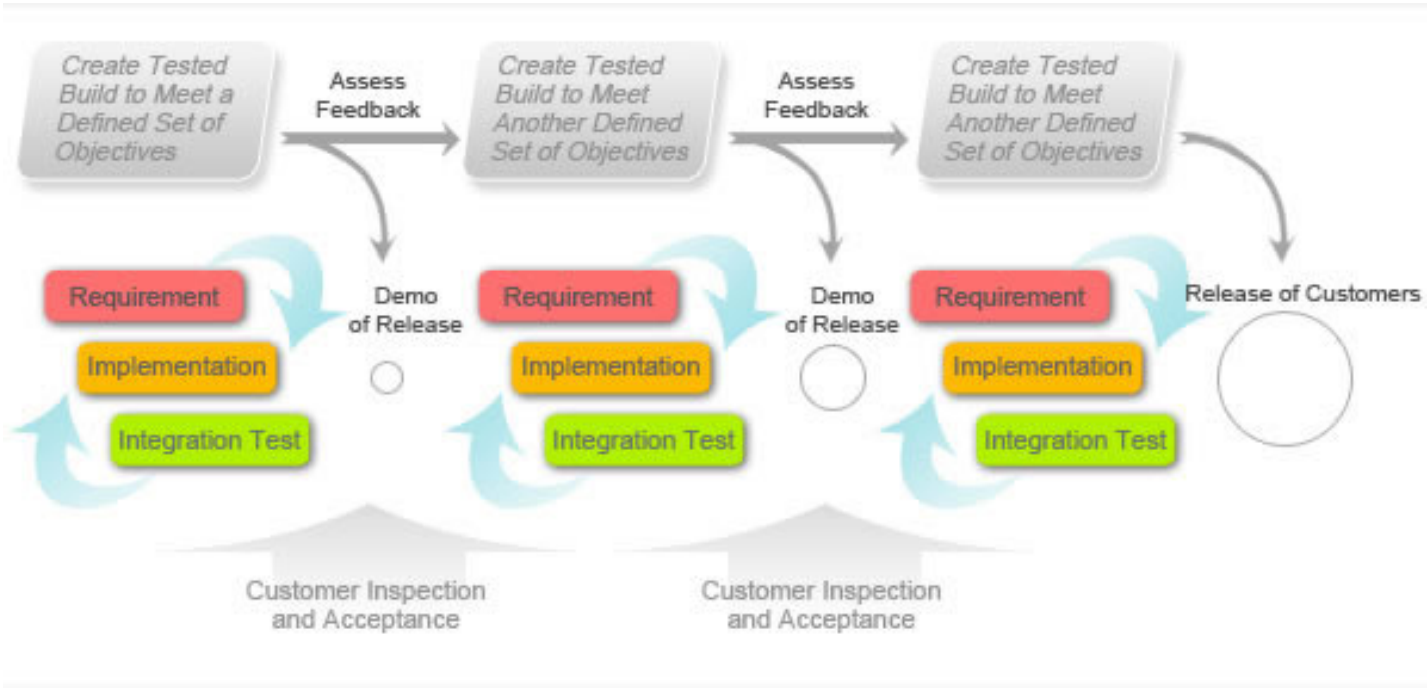
Potential Supply Chain 2



The Traditional (Waterfall) SW Development Lifecycle



Agile Development Lifecycle

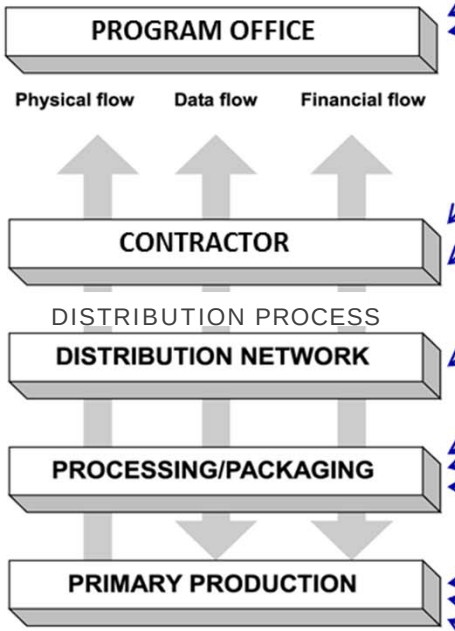


http://www.agilegator.com/pmdevelopment.html

Generic Threats – Supply Chain Attacks

Representative attacks illustrate where in the supply chain the infiltration occurs and what the malicious insertion accomplishes

Supply Chain



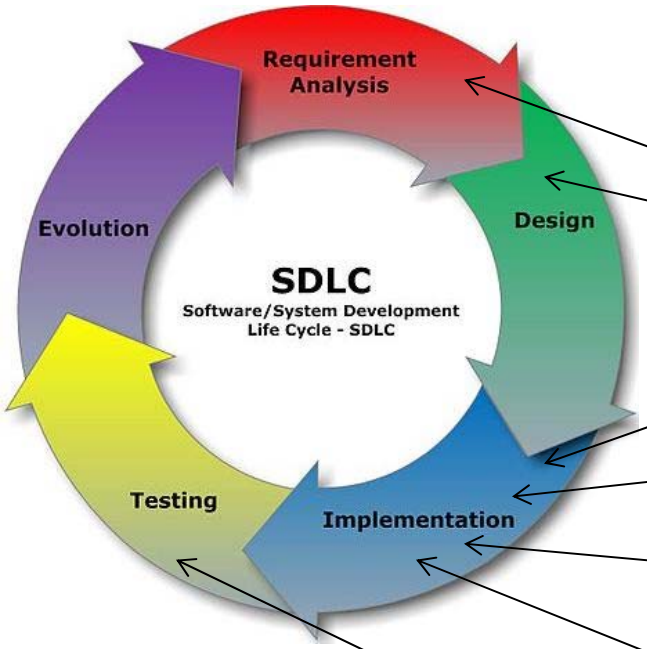
Representative Supply Chain Attacks

- Clandestine changes to mission data
- Infiltration of sites to insert back doors and malicious logic into some micro electronics (FPGAs and other devices)
- Infiltration of company receiving department to add / substitute components with backdoors to allow remote penetration during operations, denial of service, etc.
- Infiltration of transportation companies to intercept DoD component shipments (developmental or COTS) and substitute components that have malicious code inserted
- Insertion of malicious software in the open source used for math libraries
- Infiltration allowing malicious software implantation through 3rd party bundling
- Establishment of shell company to insert counterfeit parts
- Infiltration to manipulate the hardware or software baselines
- Infiltration of company software development to insert software which exfiltrates data
- Infiltration to compromise the design/fabrication of hardware

Can have multiple levels: OEMs → subassembly suppliers → assembly suppliers → integrators

Generic Threats – Malicious Insertion in the Software Development Life Cycle

Representative attacks illustrate what part of the SDLC is targeted and how malicious insertion is accomplished



Attack Vectors for Malicious Code Insertion

- Hidden in software's design (or even requirements)
- Appended to legitimate software code
- Added to linked library functions
- Added to installation programs, plug-ins, device drivers, or other support programs
- Integrated into development tools (e.g., compiler generates malicious code)
- Inserted via tools during system test

Generic Threats – Malicious System Exploitation Attacks

Representative Attacks and Vectors for Malicious Exploitation of Fielded Systems

Configuration, Operational Practices
Supply Chain (penetration, corruption)
Malware (downloaded, embedded)
External Mission Load Compromise
DNS Based Threats (cache poisoning)
Applications (built-in malware)
E-mail Based Threats (attachments)
Data Leakage (via social media)
Password Misuse (sharing)

Denial of Service (embedded malware)
Kill Switch Activation (embedded malware)
Mission Critical Function Alteration (embedded malware)
Exfiltration (by adversary)
Network Threat Activity (host discovery)
Compromised Server Attacks (on clients)
Malicious Activity (disruption, destruction)
Auditing Circumvention (evading detection)
Web Based Threats (disclosing sensitive info)
Zero Day Vectors (vulnerabilities without fixes)
Improper File/Folder Access (misconfiguration)

- **Supply Chain**
- **Embedded Malware**