



DRAPER

# A JOINT EFFORT TO ENABLE CYBER SECURITY

**Joe Brule and Mike Ridge**

OCT 3, 2018

# Agenda

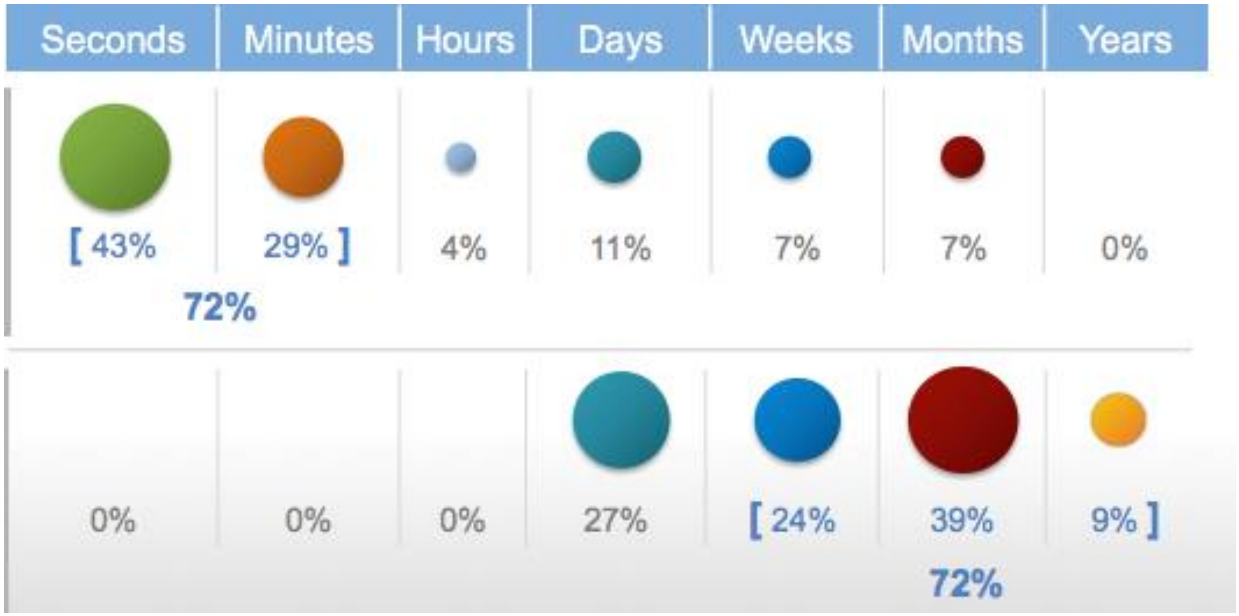


- Overview
  - OpenC2
  - JCAUS
- Prototype Implementations
  - OpenC2 Efforts
  - JCAUS Efforts
  - Joint Effort
- Findings
- Way forward for JCAUS/OpenC2

# So How's It Working Out For You?



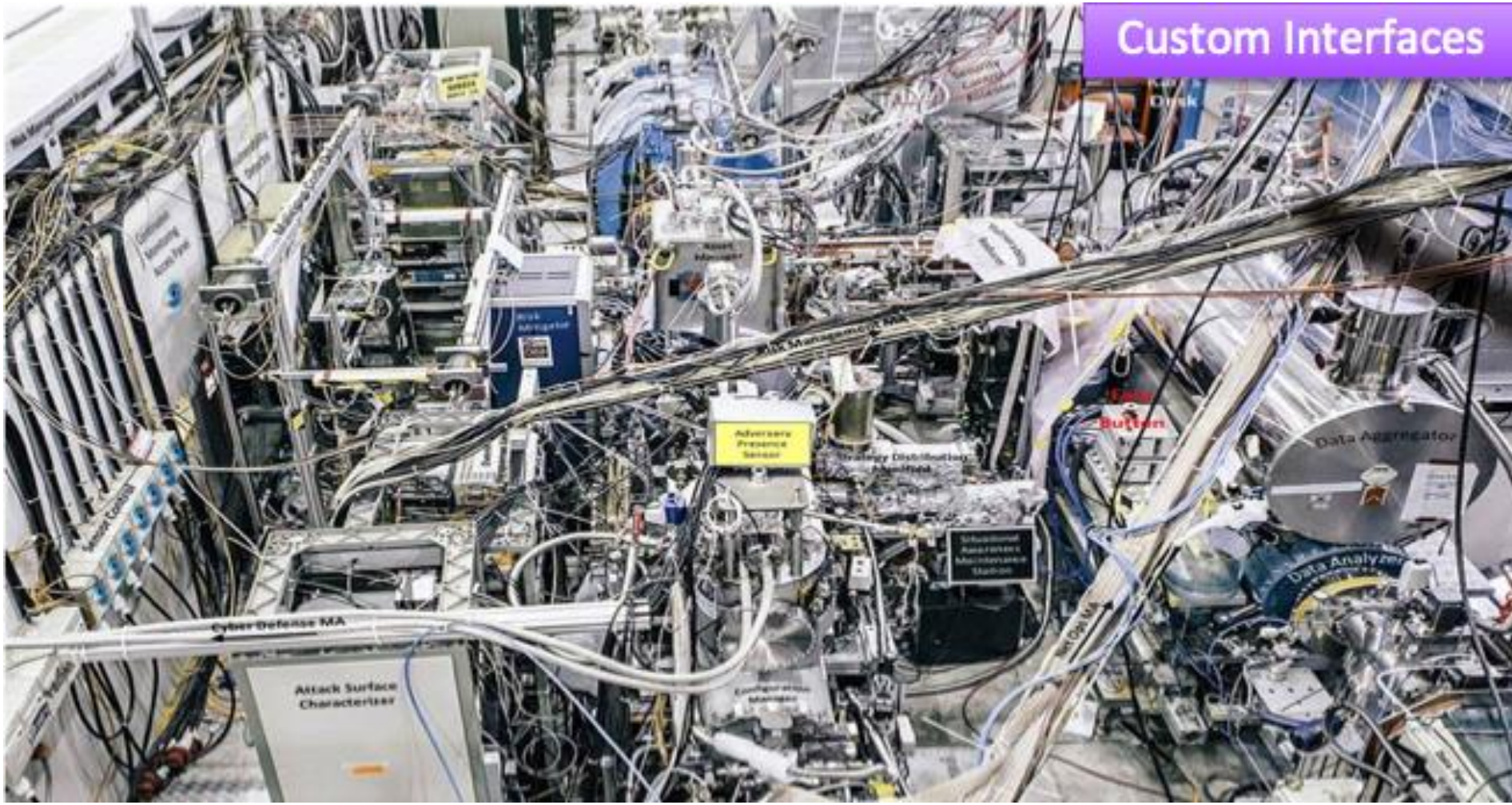
- Cyber Attacks
  - Sophisticated
    - Adaptive
    - Automated
  - Occur in Seconds
- Cyber Response
  - Slow
  - Manual




- Current State:
  - Global Attack Surface
  - Attackers Operating at Machine Speed
  - Defenders Utilizing Statically Configured Point Defenses

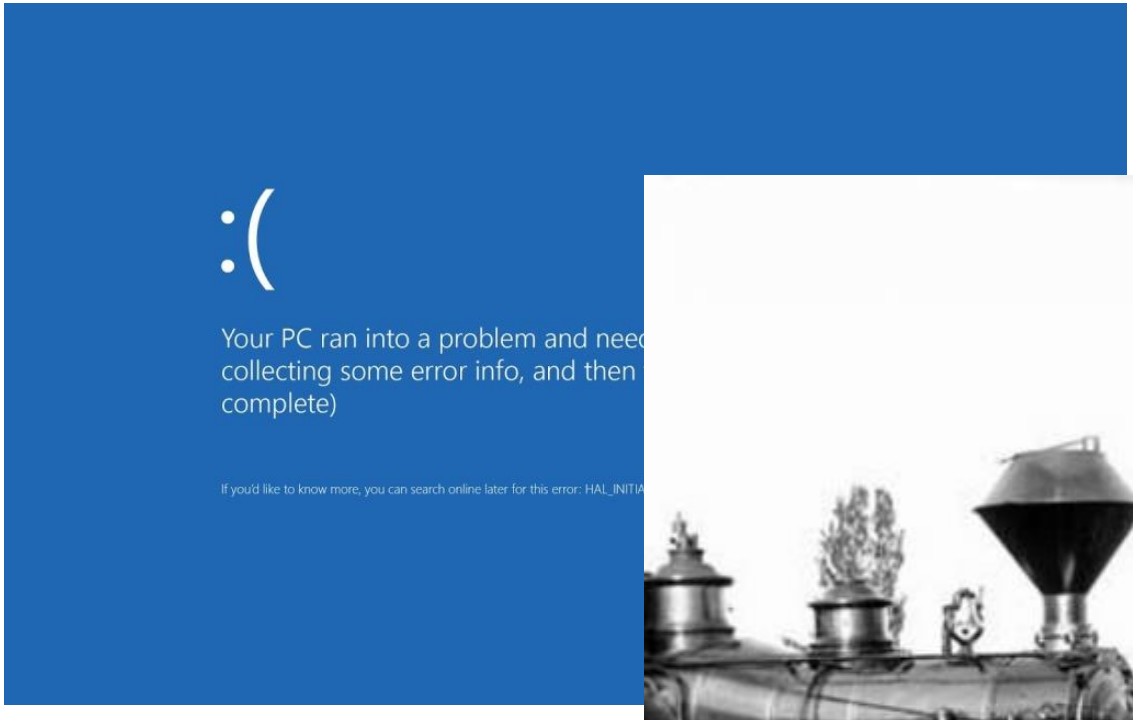
# Integration in the Absence of Standards

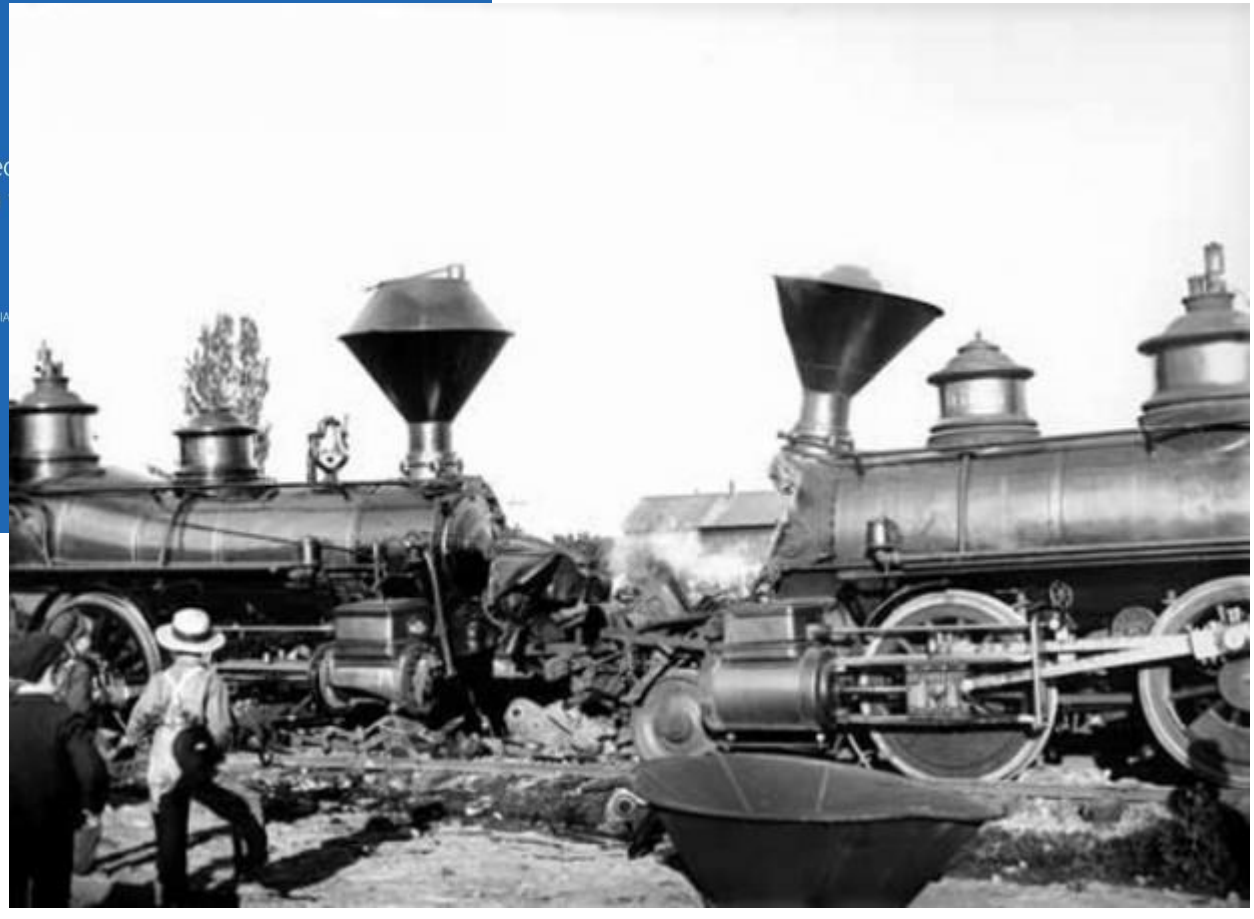




# What Can Go Wrong?





# OpenC2 is Part of a Bigger Picture




- STIX
  - Standard Threat INTEL object
  - Supports Analysis


- MQTT
  - Standard Transfer Protocol
  - Supports Pub/Sub Architecture

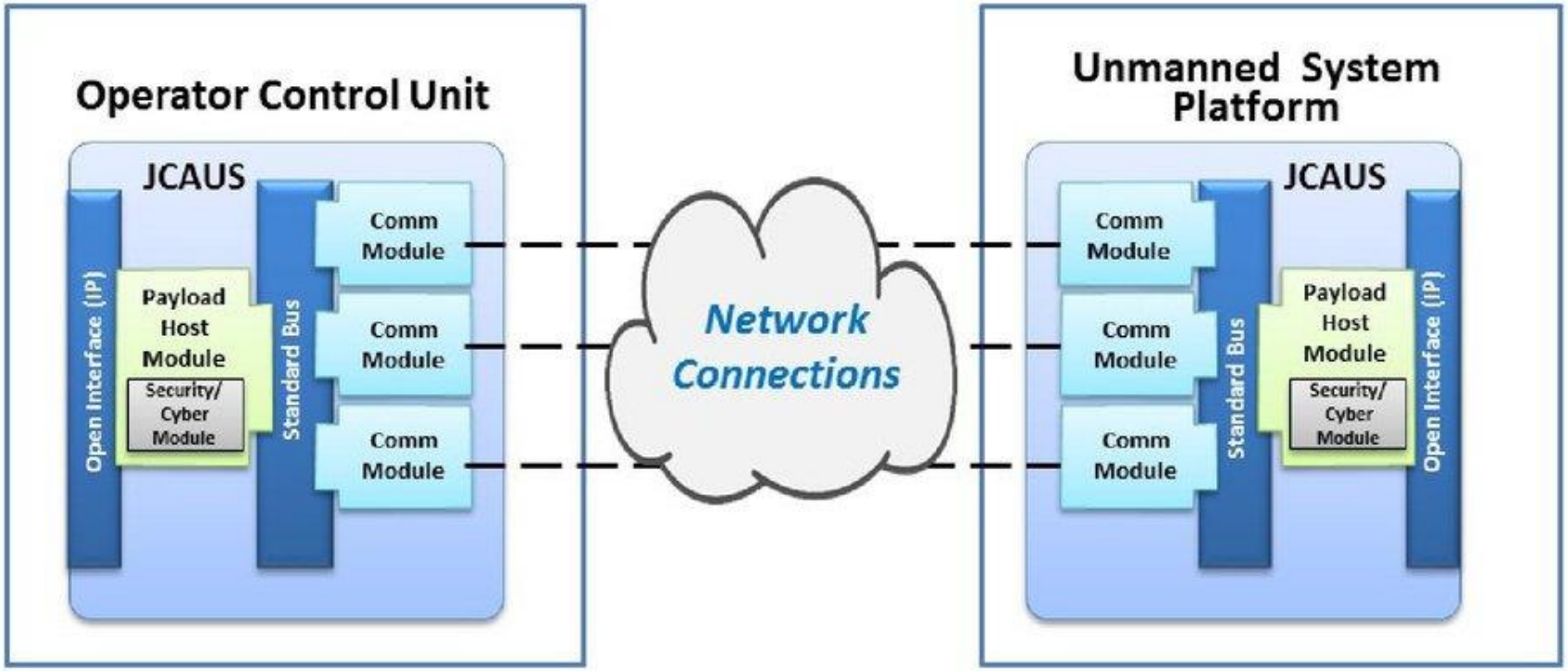

- OpenC2
  - Standard Command Language
  - Supports Acting/ Response

OpenC2 is part of a Suite of OASIS Standards

# JCAUS

## Joint Communications Architecture Unmanned Systems






**The JCAUS Architecture concept is based on industry and open standards. The JCAUS team seeks to select industry standards and adopt industry best practices to refine its exiting framework and to define an architecture**

# OpenC2 in Networks and Beyond

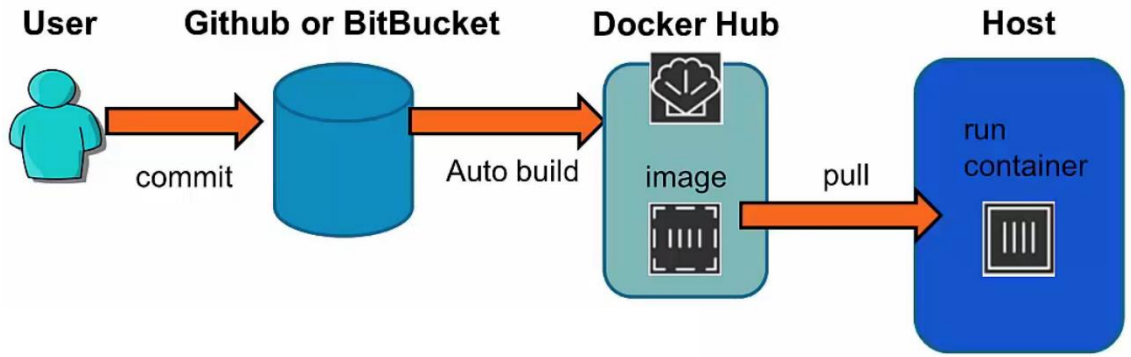


- Deny
  - Firewalls will interpret as a Rule (multiple examples)
  - Routers will interpret as ACL (Cisco CTIA)
  - Severs will interpret as permissions
- Locate
  - LYCAN use case returns GPS coordinate for an IP
- Allow
  - Mathematical compliment for Deny

What will Unmanned Platforms do with these?

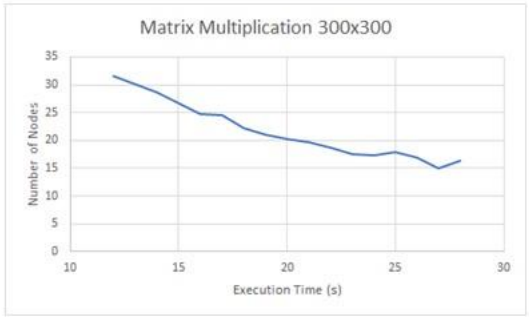
# Distributed computing w/ Docker





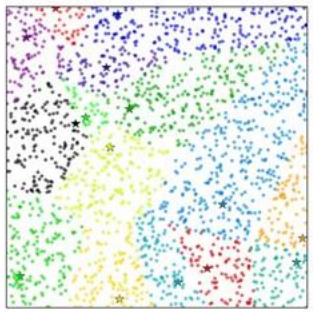

*DANIELLE SWANSON*

## K Means Parallelization




Increasing the number of nodes from 12 to 28 decreased execution time by over 48%

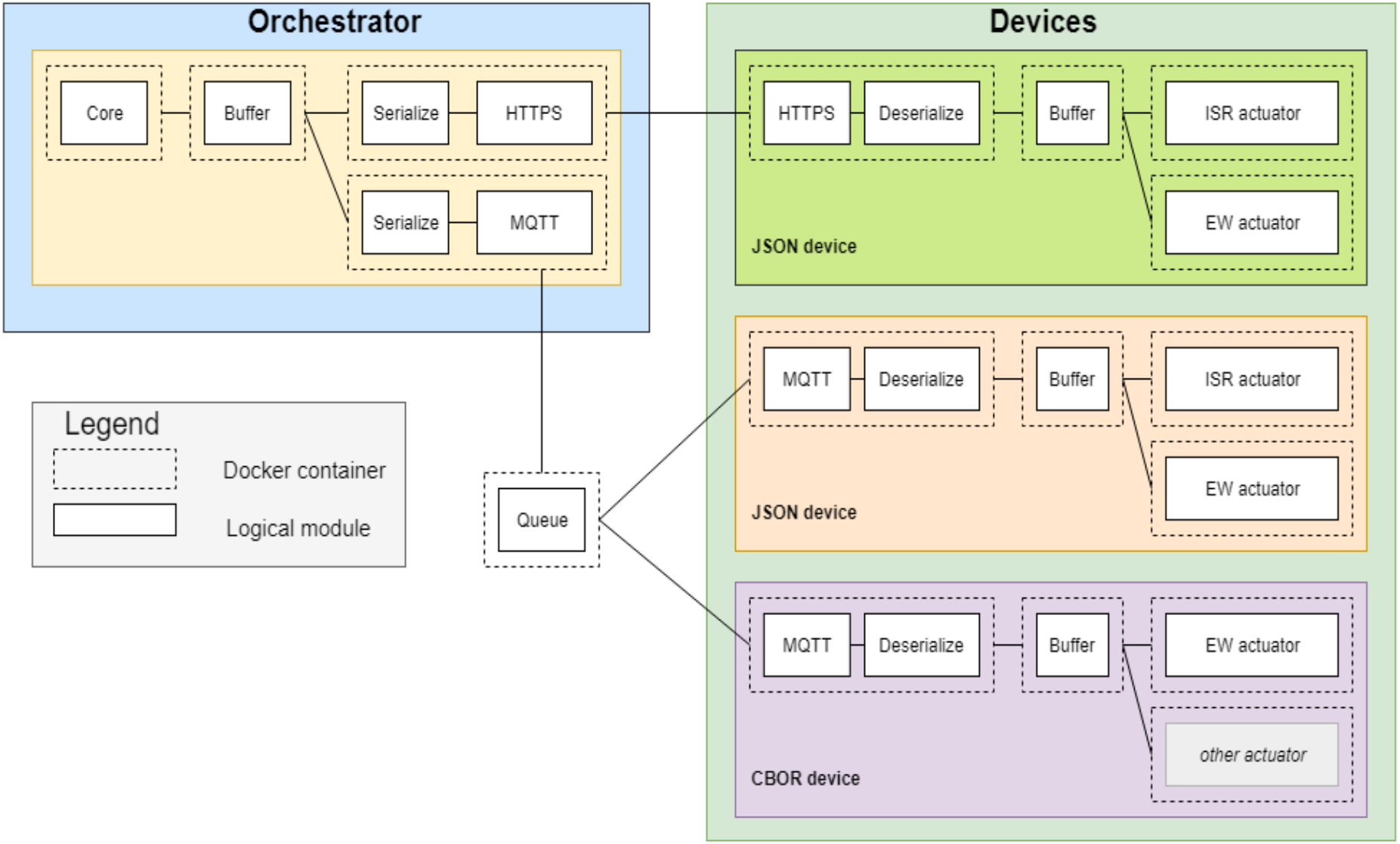
Minimize $\sum_{k=1}^{K} \sum_{x_n \in C_k} ||x_n - \mu_k||^2$ with respect to $C_k, \mu_k$



$$C_k = \{x_n : ||x_n - \mu_k|| \leq \text{ all } ||x_n - \mu_l||\} \quad (1)$$

$$\mu_k = \frac{1}{C_k} \sum_{x_n \in C_k} x_n \quad (2)$$

# OpenC2 Implementation for JCAUS



# Existing Universal Controller Requirements & Architecture

Find: CRS — Previous — Next

Distribution A: Approved for Public Release

RQ-11 Raven (MRR)

Short Range Recon (SRR)

RQ-20 Puma (LRR)

Lethal Maneuver Aerial Munition System (LMAMS)

PM UAS

PM UAS

PM UAS

PEO Missiles & Space

**Architecture Extensible to RAS ICD Systems**

PD-100 / Soldier Borne Sensor (SBS)

IOP/JAUS

PEO Soldier

SMET

SPAWAR MOCU 3.0/4.0

OSD Unmanned Control Segment (UCS)

OSD Joint Communication Architecture for UxS (JCAUS)

Unmanned Systems Waveform/Common Robotic System (Communication Link) (CRS(CL))

Acronyms:
**IOP:** Robotics & Autonomous Systems, Ground (RAS-G) Interoperability Profile (IOP)
**JAUS:** Joint Architecture for Unmanned Systems (JAUS)
**MOCU:** Multi-Operator Control Unit

IOP/JAUS

IOP/JAUS

IOP/JAUS

IOP/JAUS

Light Reconnaissance Robot (CRS(LR))

CRS(H)

MTRS Inc II

MTRS MK II

CRS(I)

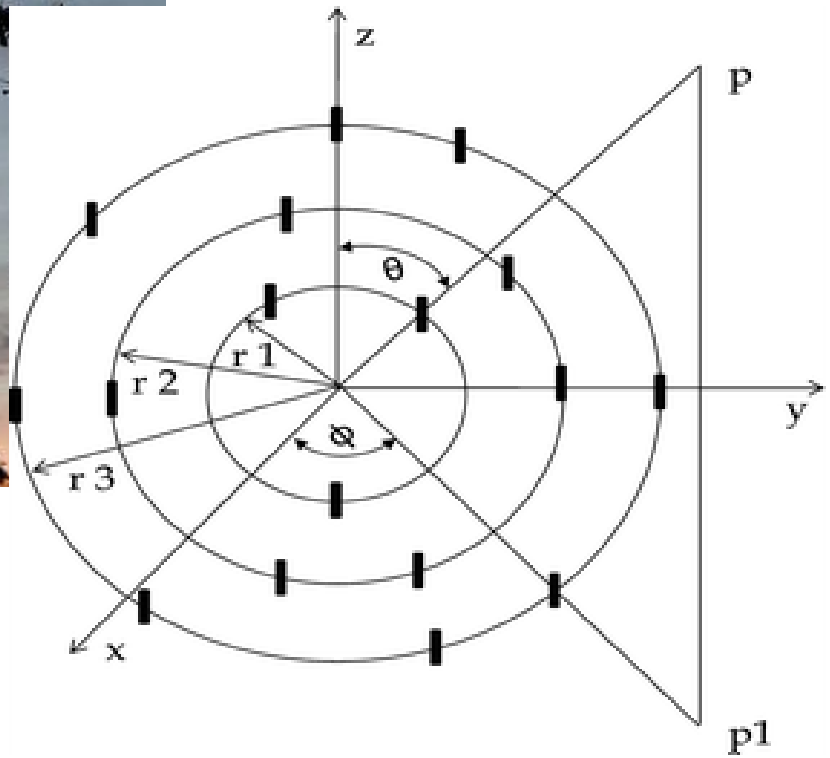
**CRS(I) Universal Controller CDD Language:** 6.2.2 (U) KPP 6 - Unmanned System Control.
The CRS (I) OCU must have the ability to achieve and maintain active and/or passive control of any current Army and Marine Corps PoR battalion and below level Unmanned (Air or Ground) System (UxS) and/or their respective payloads in less than three (T), one (O) minute(s).

CRS(I) (including Universal Controller) currently in EMD Phase

*All graphics are notional to convey the general size and type of system*

# Array of Monopole Antennas





# Maintain RF Situational Awareness



- Scenario: Concern that a burst SOI may be operating in the vicinity
  - OpenC2 commands to establish an array of antennas
    - Issue SYNC {list of identifiers}
      - Instructs UAV's to fly in formation
      - UAV's 'elect' designated router
      - Responds with Designated router
    - Issue COPY {RF range, duration}
  - End State:
    - An Array of Monopole Antennas maintaining a copy of collected signals over the past two seconds

SYNC: Synchronize a sensor or actuator with other system components

COPY: Duplicate an object, file, data flow or artifact.

# Event: Signal of Interest Identified



- Option A: Central analytic
  - Each UAS provides the TOA and Position for the SOI
  - Offline Analytic determines location
- Option B: Distribute Analytic across UAVs
  - Each formation calculates the LOB and reports

correlation matrix will be discussed in depth. By the end of this project the direction of an incoming signal will be able to be determined with the above listed methods. The system will be modeled after a uniform linear array with a varying number of elements.

## 2.Forming the Correlation Matrix

All of the methods described in this project first require calculating the correlation matrix for the antenna array and incoming signal(s) system. The process for this is described below. For a given antenna array, the array steering vector, A, is

$$A(\theta) = \begin{bmatrix} e^{ikx_1\cos\theta_1} & e^{ikx_1\cos\theta_2} & \cdots & e^{ikx_1\cos\theta_M} \\ e^{ikx_2\cos\theta_1} & e^{ikx_2\cos\theta_2} & \cdots & e^{ikx_2\cos\theta_M} \\ \vdots & \vdots & \ddots & \vdots \\ e^{ikx_J\cos\theta_1} & e^{ikx_J\cos\theta_2} & \cdots & e^{ikx_J\cos\theta_M} \end{bmatrix}$$

where J is the number of elements in the array, and M is the number of incoming signals. A gives the phase of incoming plane waves at each of the elements in the array. The signals are in turn described by a vector,

$$s(t) = \begin{bmatrix} s_1(t) & s_2(t) & \cdots & s_M(t) \end{bmatrix}^T$$

where $s_M$ correspond to each input signal. The output of the array is therefore given by

$$X(t, \theta_M) = s(t)A(\theta_M) + N(t)$$

where $N(t)$ is the vector describing the noise inherent in the system. The covariance matrices are then calculated for $X(t)$ and $N(t)$ by

# Determine Emitter Location



- OpenC2 Commands to acquire Line of Bearing
  - Issue 'SCAN {SOI}' to UAVs
    - Nodes review past two seconds and respond with TOA and coordinates
  - Issue 'LOCATE {matrix}' to ISR analytics
    - Returns coordinates of emitter
- Alternative OpenC2 Commands (low SNR environment)
  - Issue 'REPORT {SOI, LOB} to designated router
    - Issues SCAN to each UAV
    - Distributed Matrix calculations for n-channel DF
    - Designated router returns LOB and coordinates of origin

SCAN: Systematic examination of some aspect of the entity or its environment in order to obtain information.

LOCATE: Find an object physically, logically, functionally, or by organization.

# Other Scenarios



- **Avoid Radar Detection:** Analytics have determined that a potential adversary is using a radar signal to determine the physical location of the task force
  - Issue DENY [center freq, rule_number]
    - The UAV's emit a radar jamming signal
  - Issue DELETE [rule_number]
- **Include Other Sensors:** Unmanned platforms based from an adjacent carrier group is providing ISR
  - Allow [list identifiers] to the ingress of the draper tool providing key management
    - The TRANSEC key is provided to the peer task force

# Findings



- Maintained Separation of Concerns
- Agnostic of Topology
  - STAR utilizing HTTPS
  - Pub/sub utilizing MQTT
- Agnostic of serialization
  - JSON, CBOR
- Same ‘Strategic’ effect achieved from the commands
  - Deny [RF ] *(Jamming signal sent)*
  - Allow [asset_id] *(Provide TRANSEC key)*
  - Locate [RF signal]
    - Scan [SOI] returns TOA and GPS coordinate
    - N-channel array provides line of bearing

# Status of OpenC2 TC



- Language Specification
  - Final issues worked out at Oct 1,2 F2F
  - To be released for Public Comment October 17
- Actuator Specifications
  - StateLess Packet Filter Profile (October 4)
  - Stateful Packet Filter
  - Endpoint Remediation
  - SDN Controller
- Transfer Specifications
  - HTTP/TLS (October 4)
  - OpenDXL
  - CoAP

# Status of JCAUS



- The DoD plans to apply learnings into a series of unmanned vehicle systems
  - Air, Land, Maritime
  - Contracts moving independently with top level oversight.
  - Baked into the 2017-2042 Unmanned Systems Integrated Roadmap as part of the Open Systems Architectures.

https://www.efadrones.org/wp-content/uploads/2018/09/UAS-2018-Roadmap-1.pdf

# Collaborative Efforts



- Specific Engagements with students:
  - University of Illinois, Urbana
  - University of Massachusetts, Lowell
  - MIT
  - Northeastern
  - NYU
  - University of Pennsylvania

  - Spring project specifically with Northeastern

# Moving Forward



- Joint NSA/ Draper
  - Actuator Profiles:
    - Intelligence Surveillance and Reconnaissance
    - Electronic Warfare
  - Integration of Unmanned Platforms
  - Integration of Tactical and Strategic Key Management
- Request of Stakeholders
  - Identify Use Cases
  - Create Custom Actuator Profiles
  - Identify Message Fabric

Thank you!
Questions?

# Backup

# OpenC2 Codebases:



- Lycan Series
  - Translation of OpenC2 JSON to objects and back
  - Python, Java and BEAM
- OCAS
  - Simulator to validate and verify OpenC2 interface
- Python API's
  - OpenC2 API to accept & Convert OpenC2 commands to Python
  - Yuuki and Orchid are codebase
  - Reactor Master and Reactor Relay are Deployed
- OpenC2 Serializations
  - JSON (mandatory to implement)
  - CBOR & Protobuf
  - XML