“The Holistic Battlespace: Why the Key to Resilience for AI/ML Algorithms is to Leverage Complexity Science”.

By Dr. Joe Schaff

Autonomy & Avionics, NAVAIR / NAWCAD Mission Systems

 In: Proceedings of AAAI Symposium on the 2nd Workshop on Deep Models and Artificial Intelligence for Defense Applications: Potentials, Theories, Practices, Tools, and Risks, November 11-12, 2020, Virtual, published at http://ceur-ws.org

What is the Battlespace?

- **A multidomain operating environment, much like a natural ecosystem.**
- *To be effective*, the dominant force must leverage the environment to:
  1. Exploit the weaknesses of an adversary's environmental dependencies.
  2. Strengthen the dominant position by protecting key environmental factors.
- Currently, a battlespace consists of a heterogeneous mix of humans and machines, some with intelligent autonomous systems.
- Looking forward, the majority will be intelligent autonomy.
- Either of these will have a dependency on the judicious use of information – there will not be complete, but only limited data.
- To win, a dominant force needs to have awareness of its general objectives, the force laydown of both sides and any significant changes that may occur.
- Information can and should be communicated in a narrow channel as nature does – i.e. *stigmergy*.

Complexity Science – Roles of Scale & Emergence

- Ecosystem- or Battlespace-sized interactions will by default have unexpected (emergent) behaviors.
- Intelligent autonomous systems (or Complex Adaptive Systems = CAS) will need to rapidly learn and adapt to their dynamically changing environment. Effective learning must occur with limited experiences.
- Below is a list of some key issues with ML in general:
  1. **The need for adequate (i.e. massive) number of samples for comprehensive training.**
  2. **Long time scales for adaptive learning, partially due to massive sample size.**
  3. **Large computational resources needed for training.**
  4. **Brittleness due to lack of resilience, emergent misclassifications, and overfitting.**
- **Most of these are significantly different from human limitations. Let's look at the holistic picture to see how we can address some of these:**

Adressing the Autonomous Battlespace Problem from Both Ends

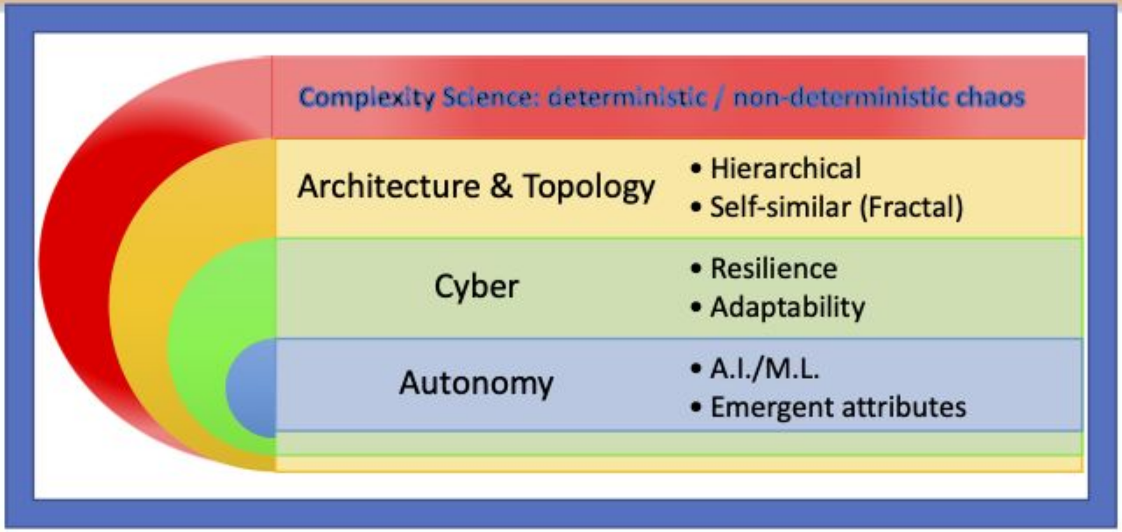
<u>A "Smart" Battlespace consists of many thousands of elements, each comprised of smart components:</u>

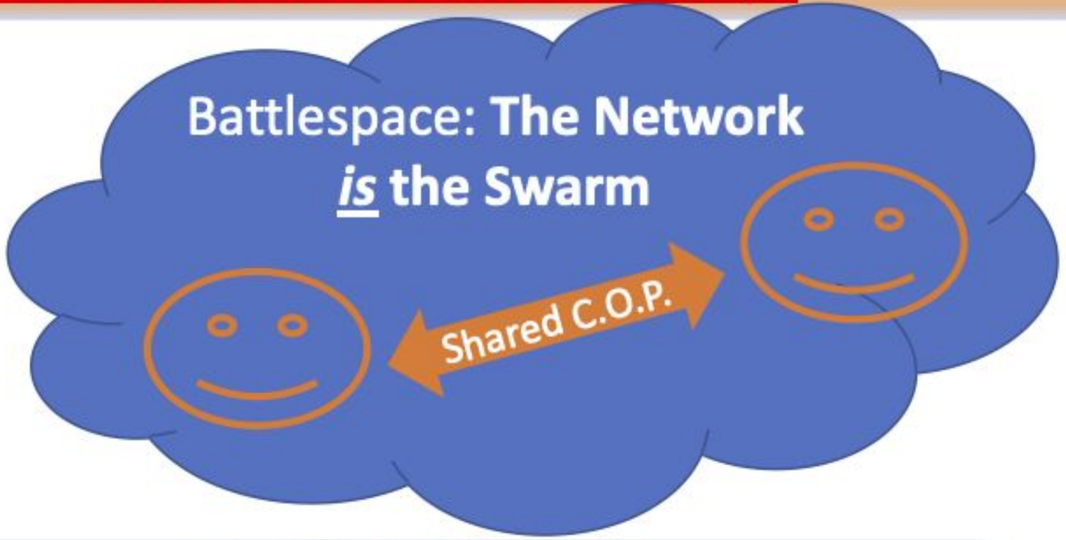
1. *Massive embedded mobile ad-hoc (MANET) radios create the "smart swarm".*
    - **<u>Both humans and machines</u>**, referred to as *"entities"* communicate = interactions.
    - Entities are *heterogeneous* and need to self-organize and be cognizant of order.
    - *<u>Mathematically equivalent</u>* problem whether you assume either radios or UAVs.
2. Entities each can consist of one or more components.
    - Components need to be resilient to attacks – i.e. self-healing and resistant.
    - Components are ***"smart components"*** that embed AI / ML to **<u>augment</u>** sensor and route planning capabilities. ***World model is the abstract "awareness".***

- ***What does ETE look like at different scales (1 & 2 above)?***

Joe Schaff, NAVAIR / NAWCAD Mission Systems

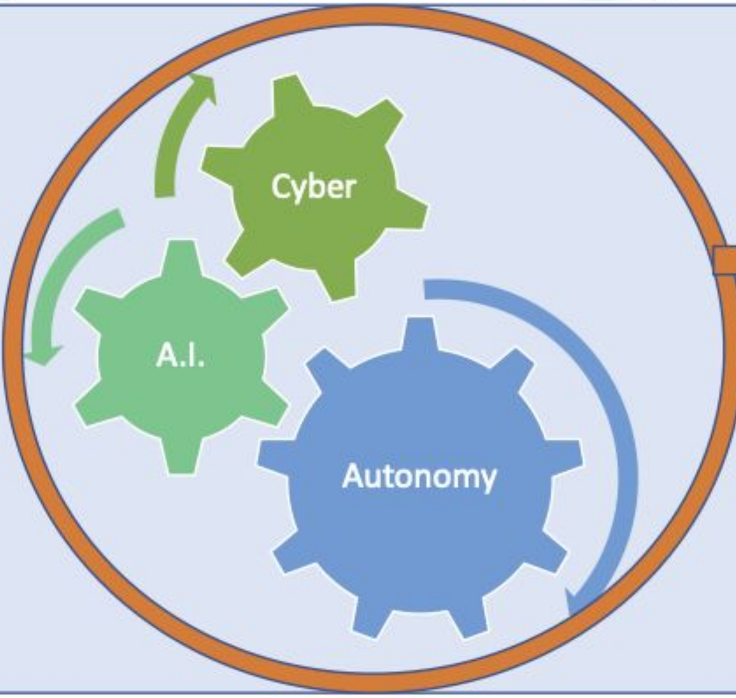
Components of End-to-End AI-enabled Autonomy:





Architecture for cyber-hardened machines to learn & adapt while creating a greater trust in their autonomous decisions

- Trust and resilience go hand-in-hand.
- Must merge Cyber and A.I. ***holistically***.
- Must allow free-reign of A.I. (i.e. creativity) but use effective resiliency constraints.
- Meta-reasoning to prevent A.I. algorithms from being deceived.



Resilience & the desired attributes of behaviors creates trust.

Joe Schaff, NAVAIR / NAWCAD Mission Systems

Complexity is Integral to Battlespace

- Battlespace by necessity **<u>must</u>** be complex.
  - Attempts to over-simplify result in easily targetable entities.
- Emergent behaviors will occur whether you want them or not.
- Best choice: “when you can’t beat ‘em, join ‘em”:
  - ***<u>leverage these behaviors to produce tactical advantages.</u>***
  - ***<u>Use these to create self-healing resilient networks.</u>***
  - ***<u>Use the “creativity” that can emerge from nonlinear classifiers in AI.</u>***
- **<u>Choose wisely</u>** where you use emergent aspects of complexity, how you apply AI.
- Constrain other systems / components as needed to make best use – e.g. formal methods.
- Be the *“lion tamer”* of complexity to gain <u>winning tactical advantages</u>.

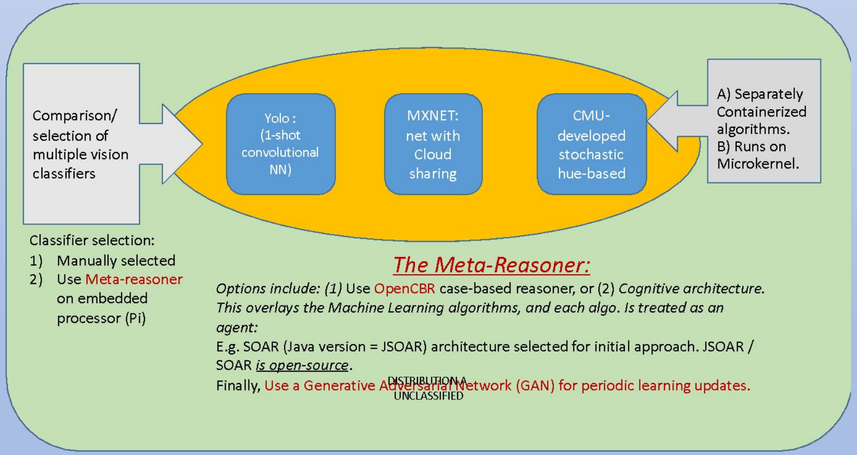
Joe Schaff, NAVAIR / NAWCAD Mission Systems

Complexity of Scale: From Swarms to Components

(*Red* = *degree of complexity being used*)

Massive Swarms → **Platforms** → **Platform Components (use of AI/ML)**

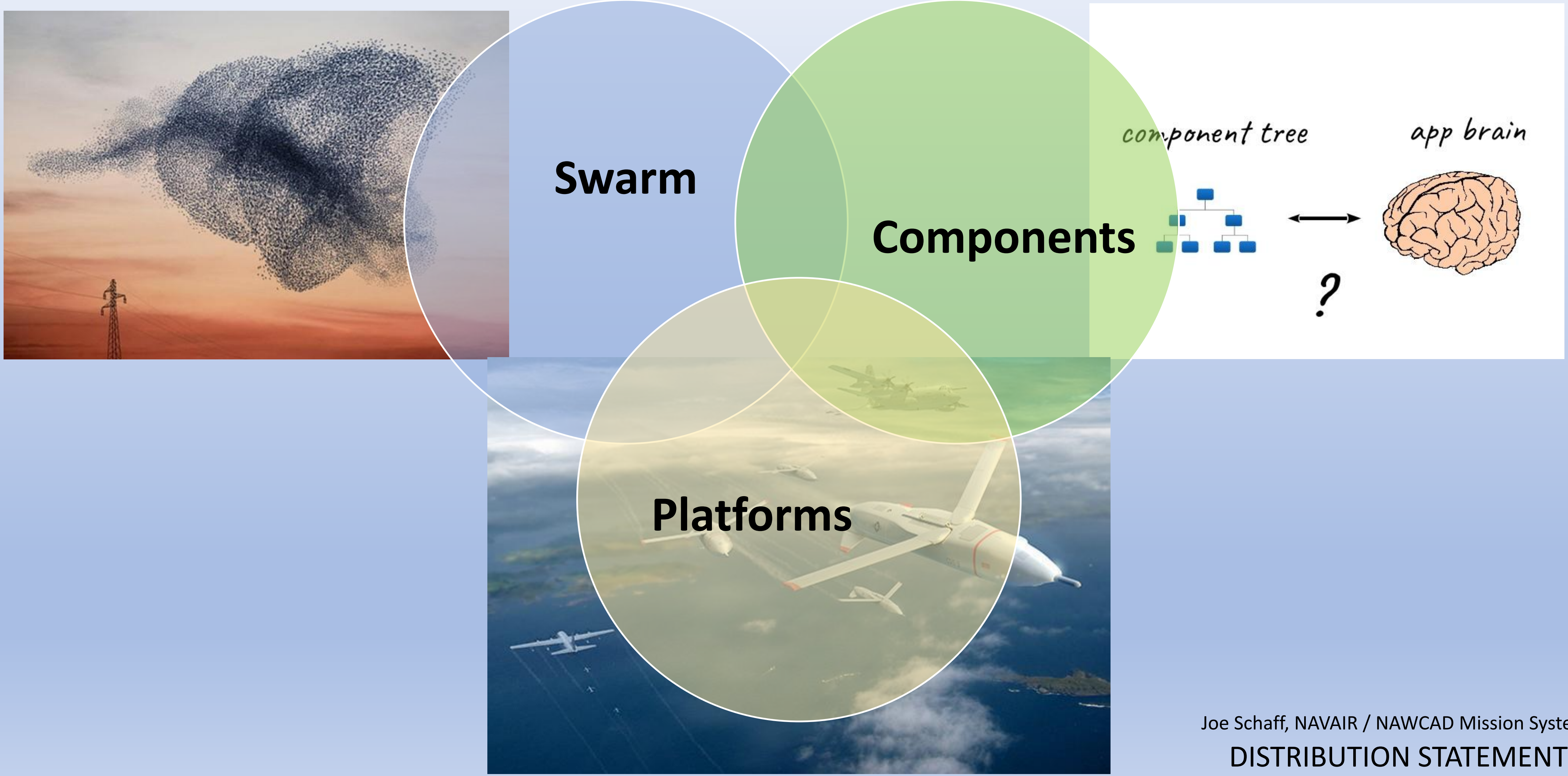
Swarm Cloud (10,000's objects)



Platform Component Architecture

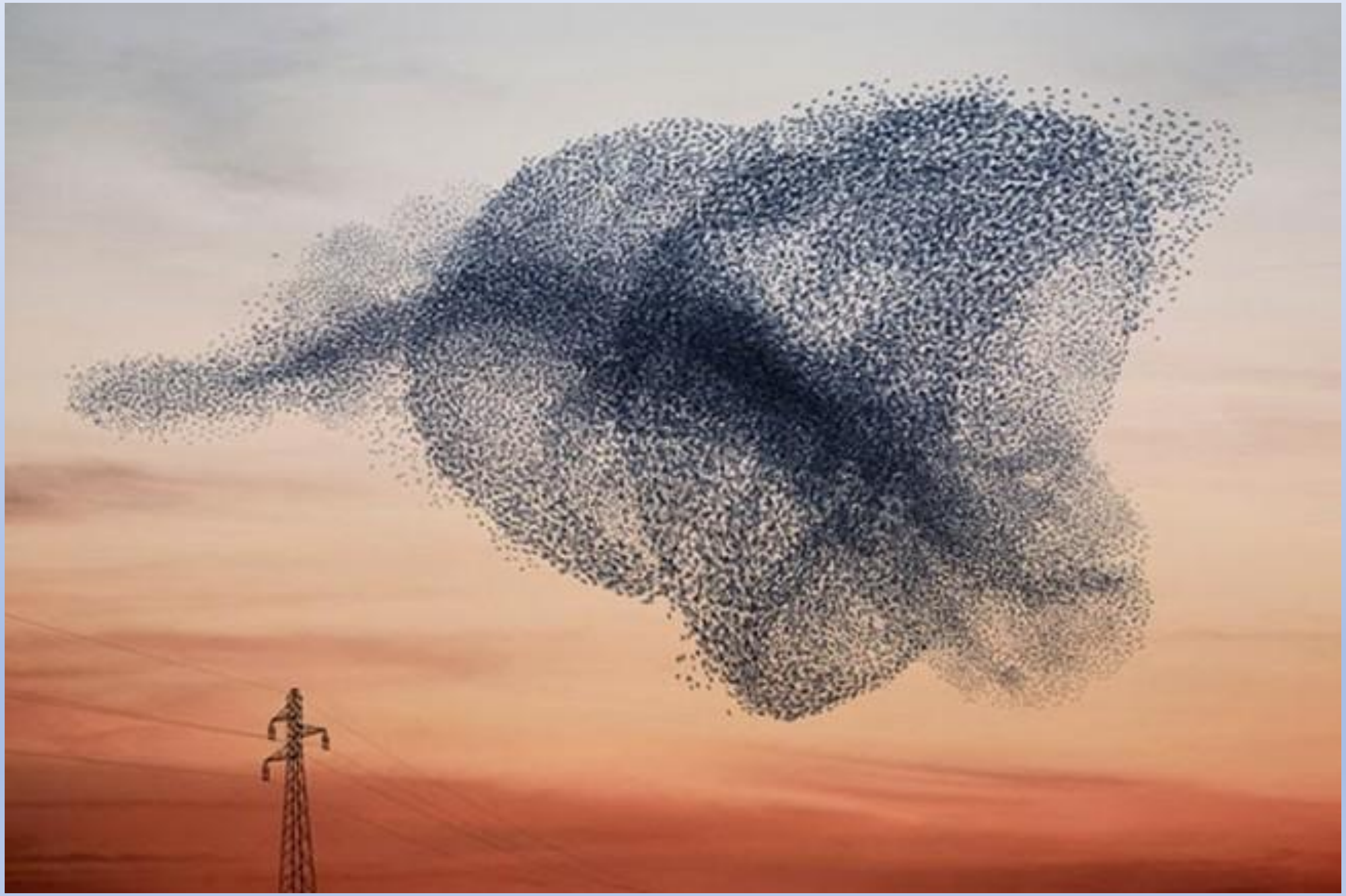
Joe Schaff, NAVAIR / NAWCAD Mission Systems

Technologies of Scale Must Overlap



Joe Schaff, NAVAIR / NAWCAD Mission Systems

<u>Technology Overlap:</u> #1 – *Massive Smart Swarm:*
Self-organizing mathematics = uses "deterministic chaos"



Joe Schaff, NAVAIR / NAWCAD Mission Systems

From Random to Order

Video: https://youtu.be/iggsygNPEnU

Joe Schaff, NAVAIR / NAWCAD Mission Systems

How Can This Possibly Work?

- Randomly generated, but constrained topology.
- Does translation / rotation (mathematically = ***affine transformation***).
- Implicitly self-similar.
- Computationally simple math
  - iterations (Iterated Function System = IFS).
  - In this particular function only one float multiplication per iteration: e.g. for determining the topological layout of 10,000 entities, would be 10KFLOPs.
  - Any IoT / edge device would have computational power to get topological picture of battlespace / other in milliseconds or faster (e.g. ESP32 = 400µsec).
- **So, what do we do with this? Distributed C2 / Resilient comms in denied environments? Control massive swarms?**

How

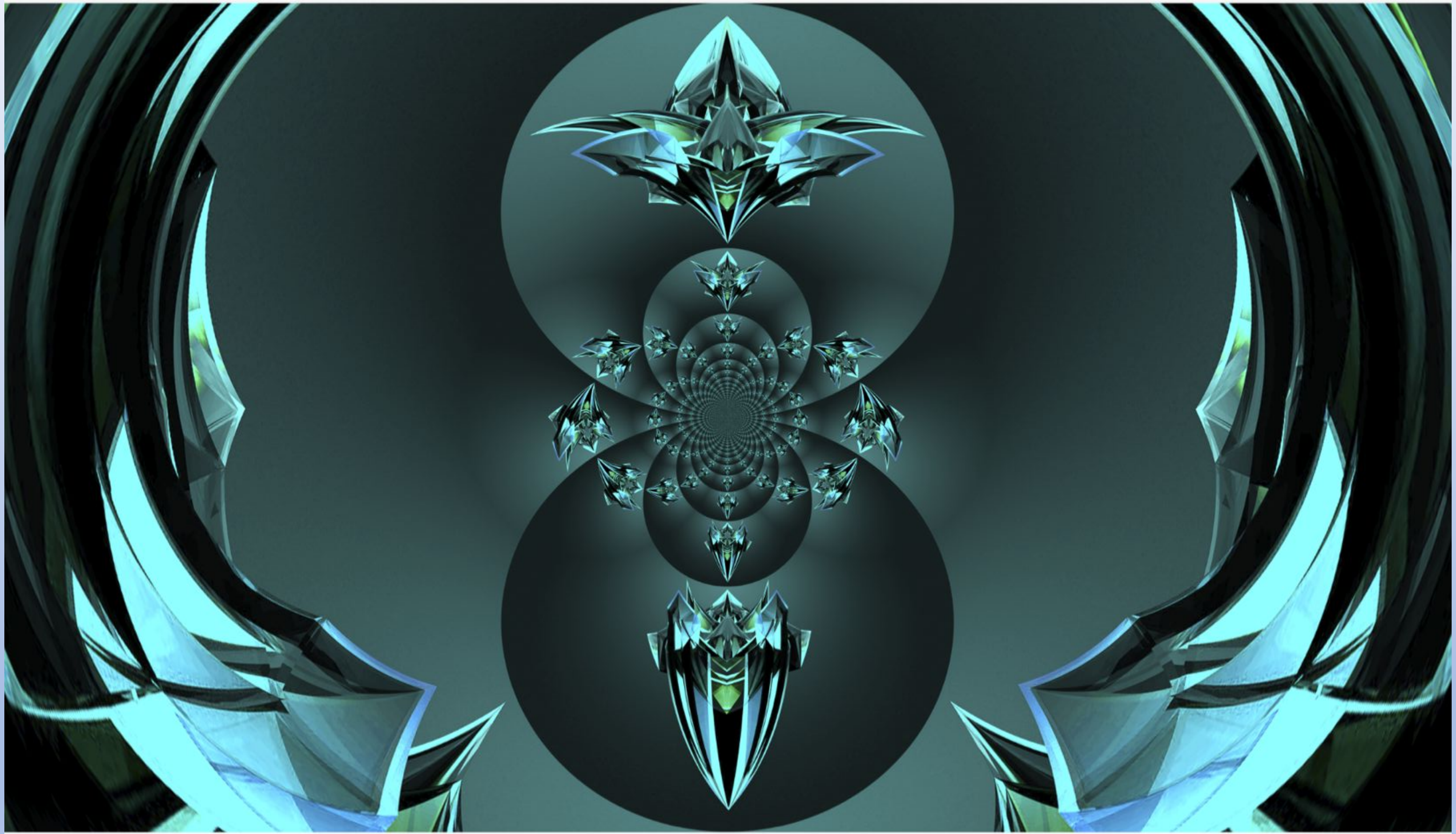
Human Immersion into Battlespace:

1) Put on Oculus / other headset
2) Link controls (BCI / other) to one of the UxVs in proximity circle.
3) Pass token to first one to respond / arbitrary choice.
4) View what it “sees”, and fly in its “world”.
5) Handoff token when done / other location needed.

OK, but what is it???

Joe Schaff, NAVAIR / NAWCAD Mission Systems

It’s a Fractal!



Further details can be found in the chapter I wrote (Leveraging Deterministic Chaos to Mitigate Combinatorial Explosions) for the book “Engineering Emergence: A Modeling and Simulation Approach”, CRC Press ©2019.

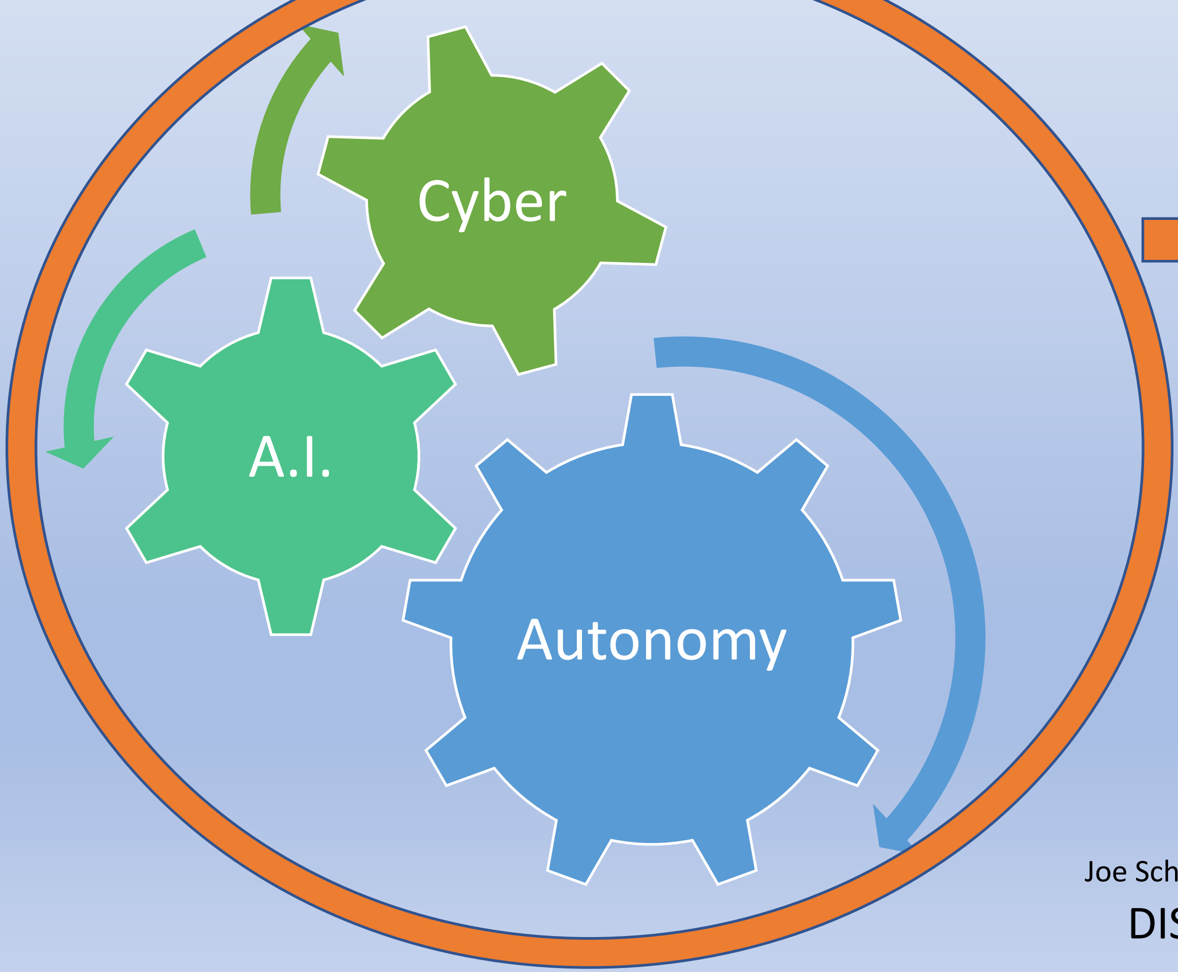
Joe Schaff, NAVAIR / NAWCAD Mission Systems

DISTRIBUTION STATEMENT A

<u>Technology Overlap</u> *#2 Components - Resolving Trust:*

Architecture for cyber-hardened smart components to learn & adapt while creating a greater trust in their autonomous decisions

- Trust and resilience go hand-in-hand.
- Must merge Cyber and A.I. ***holistically***.
- ***Must allow free-reign of A.I. (i.e. creativity) but use effective resiliency constraints.***
- Meta-reasoning to prevent A.I. algorithms from being deceived.



Resilience & the desired attributes of behaviors creates trust.

Patent disclosure was submitted and presented to Invention Evaluation Board

Joe Schaff, NAVAIR / NAWCAD Mission Systems

Adversarial AI: Natural Adversarial Examples*

- Natural adversarial examples from IMAGENET-A. The red text is a ResNet-50 prediction with its confidence, and the black text is the actual class.

* from: arXiv:1907.07174v2 [cs.LG] 18 Jul 2019

How do we avoid some of these issues?

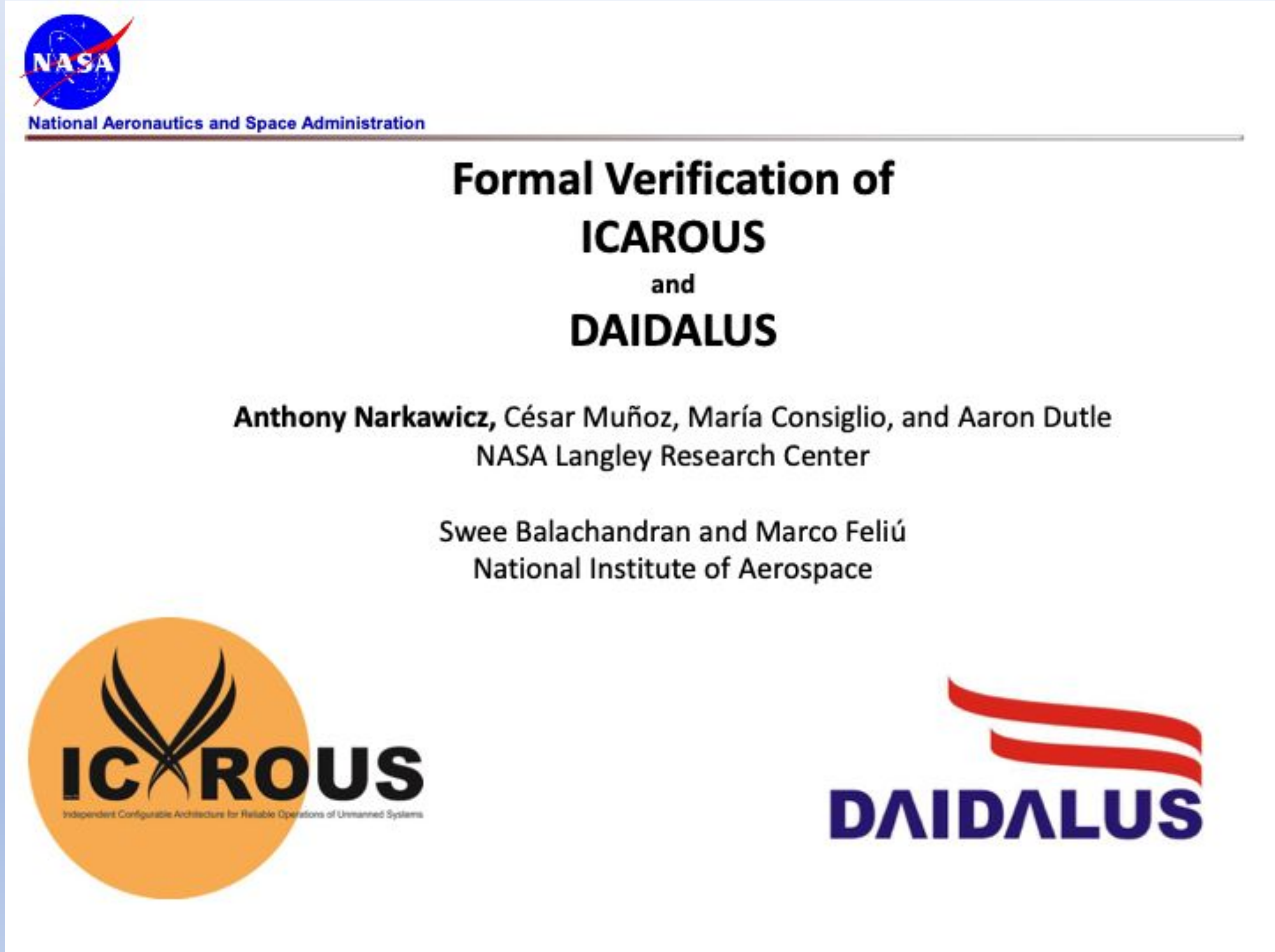
- We may never be able to design “foolproof” resilience into a system.
- There are good strategies to limit some of the weaknesses in AI/ML.
- Some aspects of transfer learning – IF the data is “clean” to begin with: “An Empirical Evaluation of Adversarial Robustness under Transfer Learning”
- Others may not be avoidable if data is “poisoned”. See: Poison frogs.
- ***First steps: Architecting trustworthy resilience and validating these architectures***

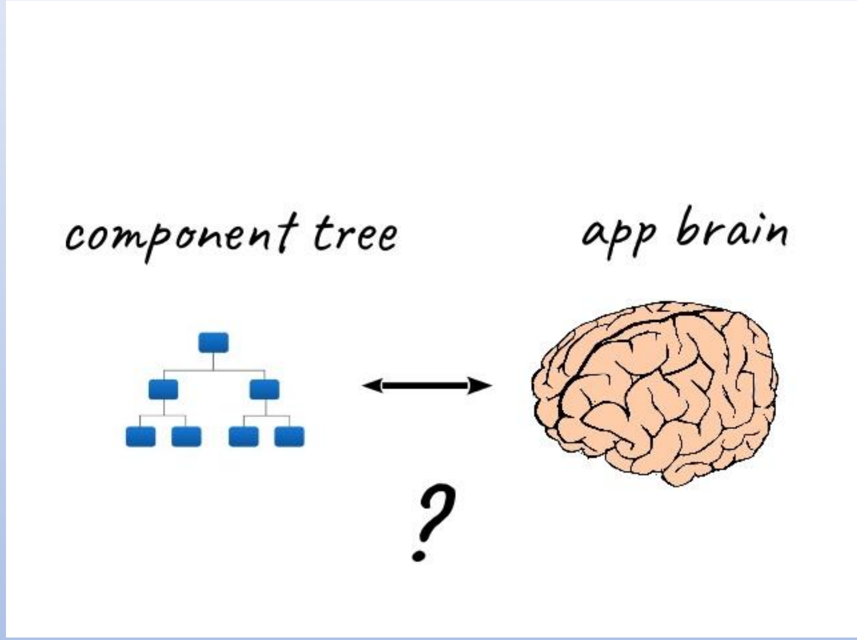
Architecting Resilience – some "Puzzle Pieces"

DARPA started the Assured Autonomy program.

- ***This program looks at the methods for some AI / ML validation, but does not look at the battlespace "Big Picture".***
- Early stage - Focused on AI/ML specifically.
- Funding academic research for verifying /validating performance aspects of primarily NNs
- Example:
  - VerifAI/SCENIC = toolkit for design/analysis of AI systems (SCENIC=probabilistic programming language). D. Fremont, et.al, UCal Berkeley.
  - Study uses Grand Theft Auto 5 (GTA5).
  - Download software here: https://github.com/BerkeleyLearnVerify/VerifAI
- Many more examples available from other schools.
- ***Formal Methods Approaches are frequently used.***

Formal Methods for Trust(?)…*but it doesn't Scale well*…





Can it work with "smart components"?

- *Sometimes- complexity may rule it out*

Component Architecture Background

*Architectures have been designed in the past that address some but not all of these. Below are some of the attributes of the proposed architectural approach:

- 1. Is able to use heterogeneous AI/ML technologies.
- 2. Mitigates shortfalls in specific vision/other algorithms.
- 3. Does meta-reasoning (cognitive architecture).
- 4. Is Cyber-resilient.
- 5. Is fully scalable from low-cost expendable to high value platform.
- 6. Has a fully open architecture in hardware and software.
- 7. Allows exploration of algorithm internals for AI/ML and cyber analysis.

- *** Note: “architecture” is clearly an overloaded word - if you don’t like the word “architecture”, replace it with “framework”.**

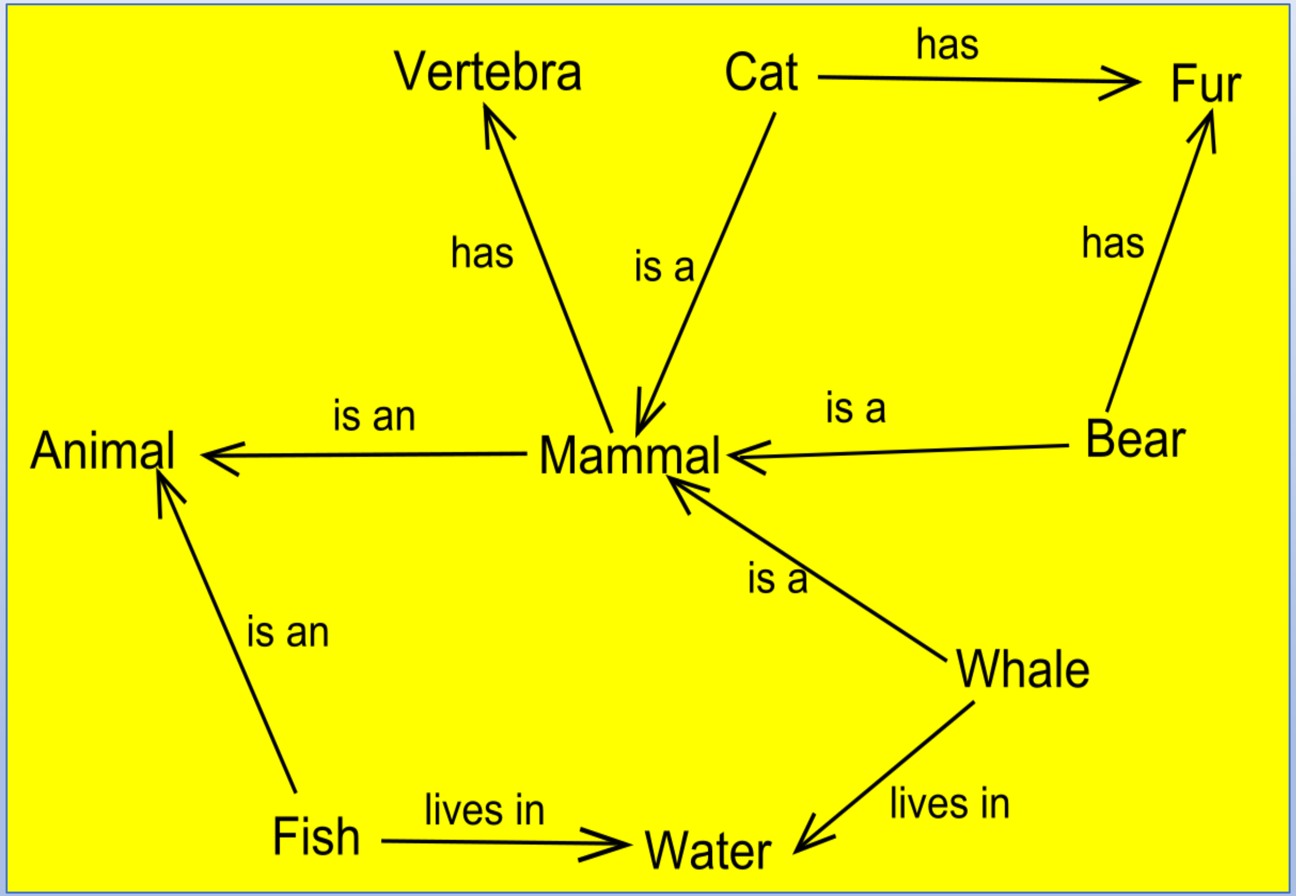
Joe Schaff, NAVAIR / NAWCAD Mission Systems

Quick Fix for Minimal Data and "*basic*" Rapid Learning

- What about DLNN issues:
  1. **adequate (i.e. massive) number of samples for comprehensive training?**
  2. **Short time scale for adaptive learning?**
- ***Transfer learning:*** take the trained weights / other parameters for similar NN trained on similar problem, load into new NN.
  - ***Issues include:* is the problem domain sufficiently similar? Does this limit the item classified to only those close / exact enough to original training data (i.e. overfitting)?**
- ***Better way:*** Use **"helper"** algorithms and mathematical functions as coarse classifiers to "pre-train" the DLNN.
  - ***Helper algorithms can work in a complementary manner with algorithms that are more accurate but challenging to train / adapt.***
  - ***More than just ensemble classifiers = these are matched complementary sets. The sets can also be combined with other classifiers for an ensemble.***

Example Helper Algorithm

- Can be solved by incorporating earlier AI/ML paradigms into architecture.
- One simple example is ***semantic net:*** members of a class and attributes are shown by connected graph.



But What if We Don't Know the Categories?

A) what if we don't know categories or relationships? B) what if the problem space is nonlinear?

- ***Example #2:* Radial Basis Function (RBF) NN** is an "analogizer" = it can estimate approximately which class something fits into, *even if classes are not yet defined* (unsupervised learning = 1st stage), then follows with a few good examples (2nd stage).
- RBFs and some SVMs (Support Vector Machines) can create categories. RBF also can address many *nonlinear problems*, e.g. chaotic time series. Convergence to control dynamics or create classes to recognize can be done with < 100 examples.

Thing 1, Thing 2, and Swamp Thing

A classic complexity / chaos example is given = logistics equation.

J. Moody and C. J. Darken, "Fast learning in networks of locally tuned processing units," Neural Computation,1, 281-294 (1989).

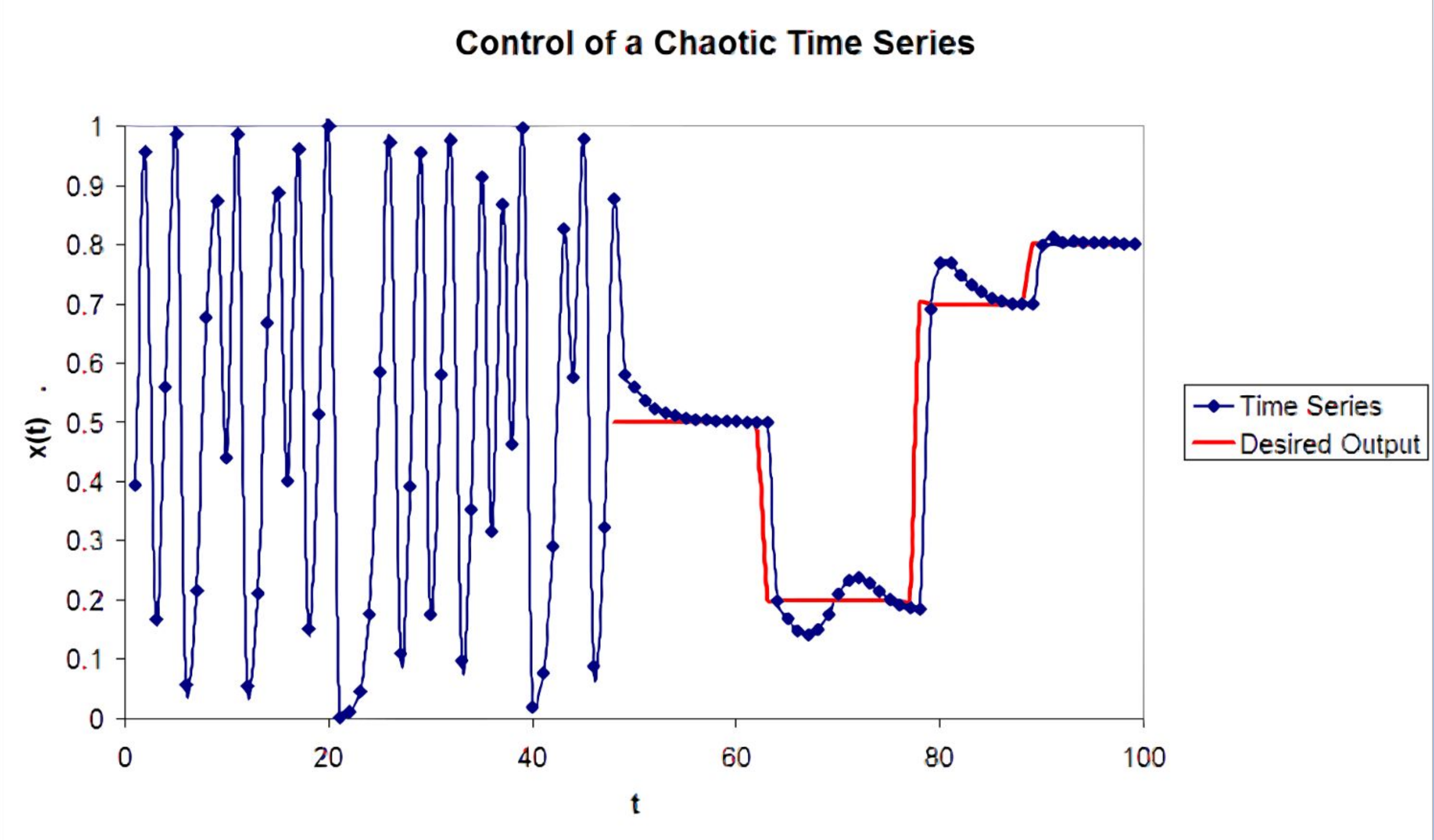
Radial basis function network: Control of the logistic map.

The system is allowed to evolve naturally for 49 time steps. At time 50 control is turned on. The desired trajectory for the time series is red. ***The system under control learns the underlying dynamics and drives the time series to the desired output*.**

Computationally simpler & faster than DLNN – just not as exact.

Permission details

This work has been released into the public domain by its author, CommodiCast at English Wikipedia. This applies worldwide. In some countries this may not be legally possible; if so: CommodiCast grants anyone the right to use this work for any purpose, without any conditions, unless such conditions are required by law.



Joe Schaff, NAVAIR / NAWCAD Mission Systems

Helper Function (mathematical type)

- Around mid-1990s I noticed that complex, almost random behaviors of NNs had some implicit pattern but could not figure it out.
- Looked at weights before, during, after training. Noticed self-similar pattern (fractal) for adjacent weights and respective inputs.
- **Hypothesis:** if the fractal pattern of trained weights is saved, then transformed & applied to similar NN topologies, this will shorten the training time & data needed.
  - ***What about overfitting to exact fractal parameters?*** Solution is to use multifractal = superimpose another similar or possibly different fractal onto original (similar techniques are used for wave functions in quantum mechanics).
  - ***What if I can determine the inverse of the fractal functions?*** Superimpose that to “undo” any learning. (multifractal link: https://imagej.nih.gov/ij/plugins/fraclac/FLHelp/Multifractals.htm)

Now…put it all together to build a cyber-resilient architecture.

Running Algorithms in a Meta-reasoning Component Architecture (an example)

Comparison / selection of multiple vision classifiers

*Yolo : (1-shot convolutional NN)

MXNET: net with Cloud sharing

CMU-developed stochastic hue-based

A) Separately Containerized algorithms.
B) Runs on Microkernel.

Classifier selection:

1) Manually selected
2) Use Meta-reasoner on embedded processor (Pi)

The Meta-Reasoner:

Options include: (1) Use OpenCBR case-based reasoner, or (2) *Cognitive architecture. This overlays the Machine Learning algorithms, and each algo. Is treated as an agent:*

E.g. SOAR (Java version = JSOAR) architecture selected for initial approach. JSOAR / SOAR *is open-source*.

Finally, Use a Generative Adversarial Network (GAN) for periodic learning updates.

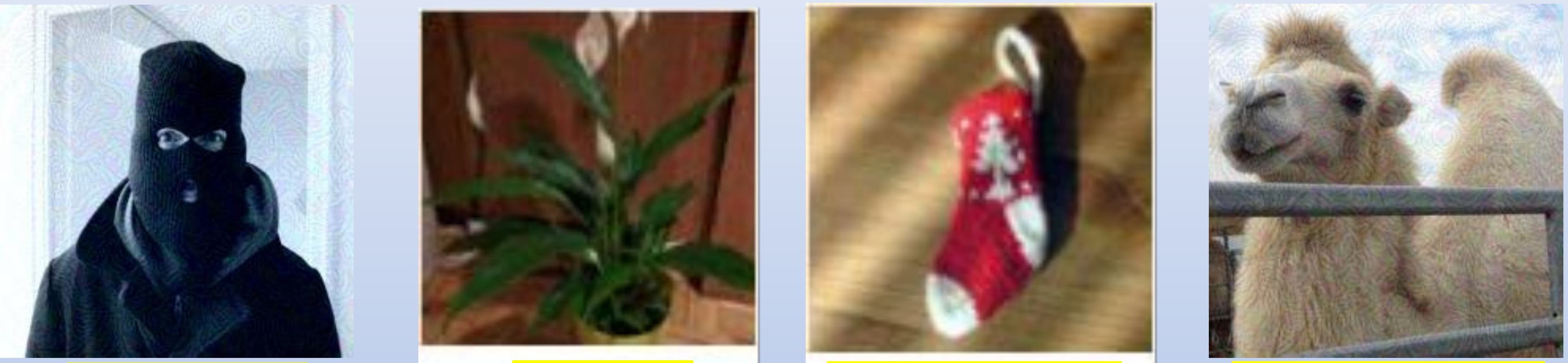
DISTRIBUTION A
UNCLASSIFIED

***Embedded Helper**

Joe Schaff, NAVAIR / NAWCAD Mission Systems

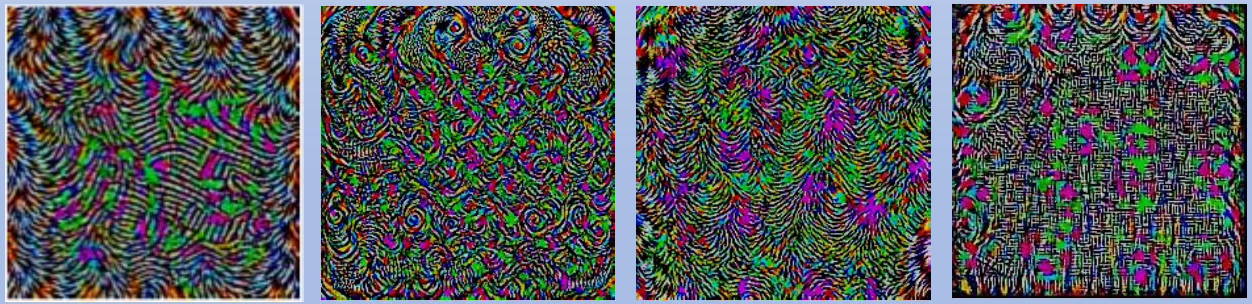
The Reason for Meta-Reasoning (“*adult supervision*”): Detecting Deep Fakes and Adversarial Perturbations[1]

***Misclassifications** (noise pattern is already embedded):*



African grey | ***Macaw*** | ***Indian elephant*** | ***Three-toed sloth***

*Some **embedded noise** patterns for different classifiers:*

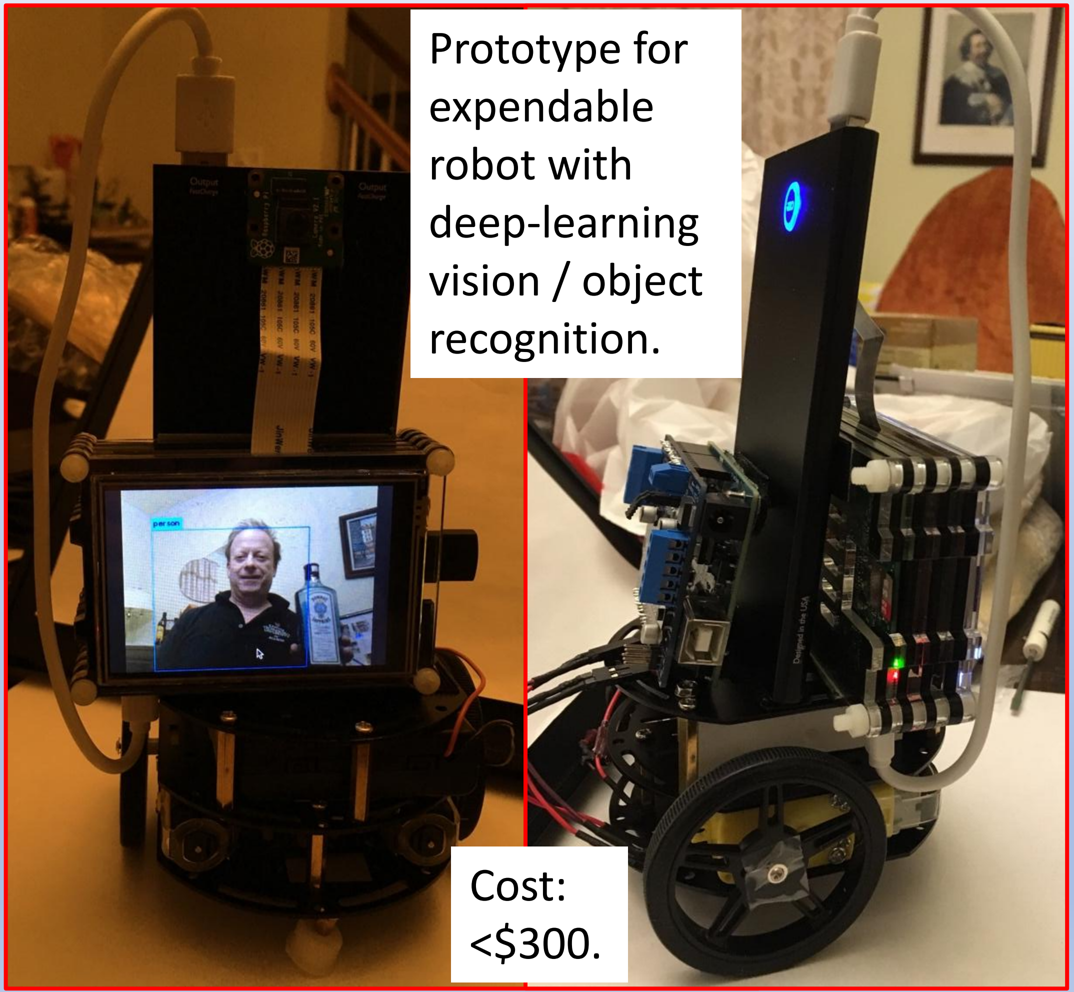


1. Extracted from: “Universal adversarial perturbations”, S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard; arXiv:1610.08401v1 [cs.CV] 26 Oct 2016.

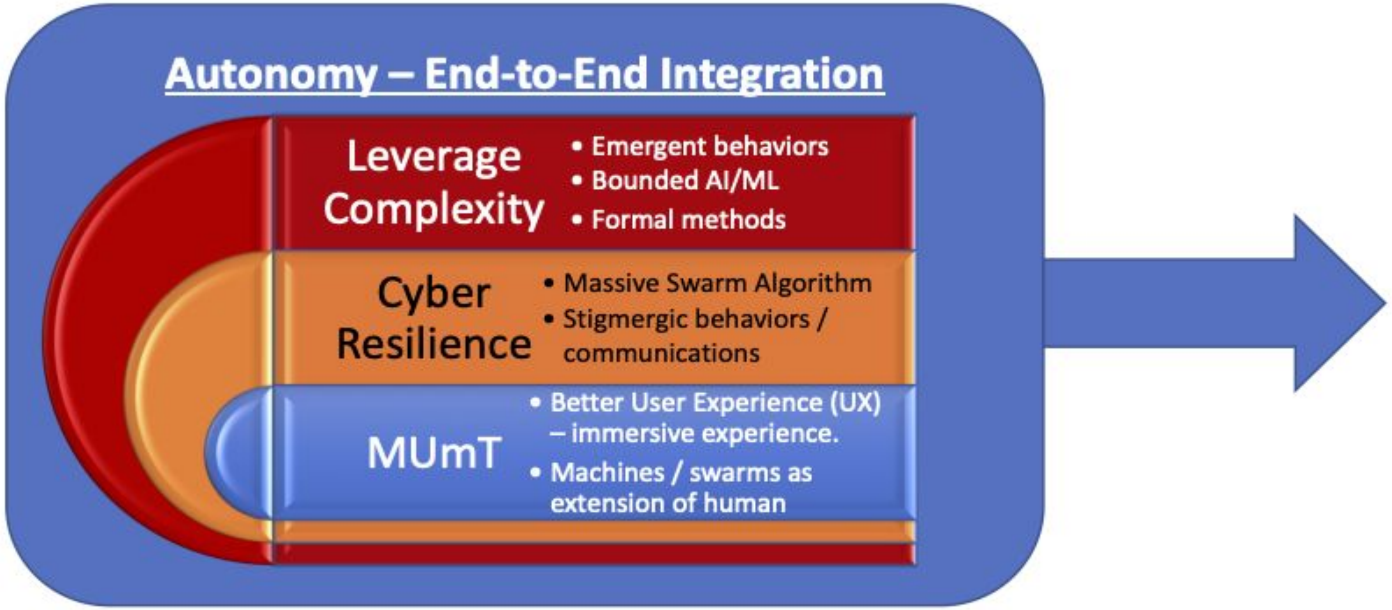
Build a Scalable Prototype for ML & Cyber, and Future Advanced Threats.

Real-time Convolutional Neural Networks for Emotion and Gender Classification (academic pub.)

Every row starting from the top corresponds respectively to the emotions {e.g. “angry”, “happy”, “sad”, “surprise”, …}
Both left & right blocks represent same pictures.
Right=convolved using backpropagation variant algorithm.



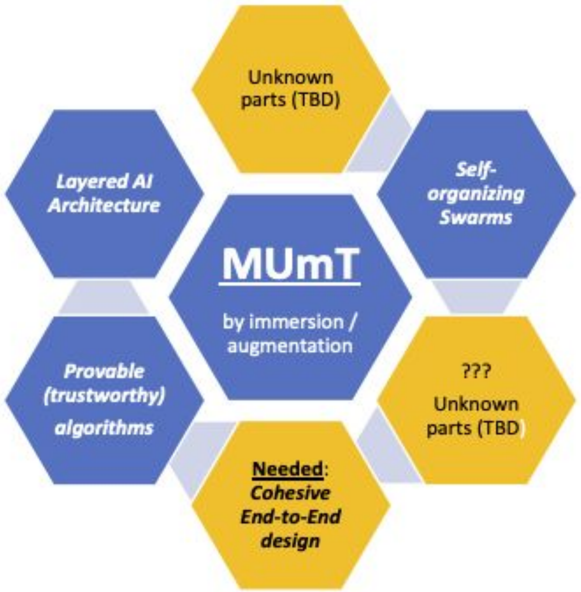
Joe Schaff, NAVAIR / NAWCAD Mission Systems

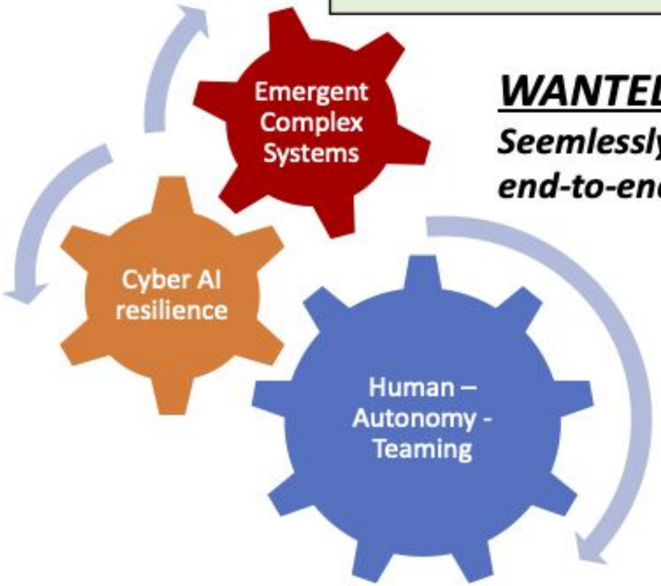


This is mostly **achievable today** – here is what we have so far:

- Algorithms for massive swarms
  - Intractable problems solved by leveraging complexity, emergent (stigmergic) behaviors.
- Cyber-resilient AI/ML architectures
  - A "meta-reasoner" layer allows <u>any</u> "creative" ML behaviors, encouraging "safe" emergence.
- Mathematically provable algorithms
  - Formal methods verified algorithms (NASA).
  - Guaranteed resilient performance envelopes.
- 3D prototype interfaces
  - Better immersion experience, control as though an extension of a human appendage.
  - Some parts needed for enhancement.

- ***We have most of the pieces (blue).***
- ***Still some parts missing (yellow).***





WANTED:

Seemlessly meshing end-to-end autonomy.

Joe Schaff, NAVAIR / NAWCAD Mission Systems

DISTRIBUTION STATEMENT A

From Prototype to Production: Overlaying a Technology Transition Architecture

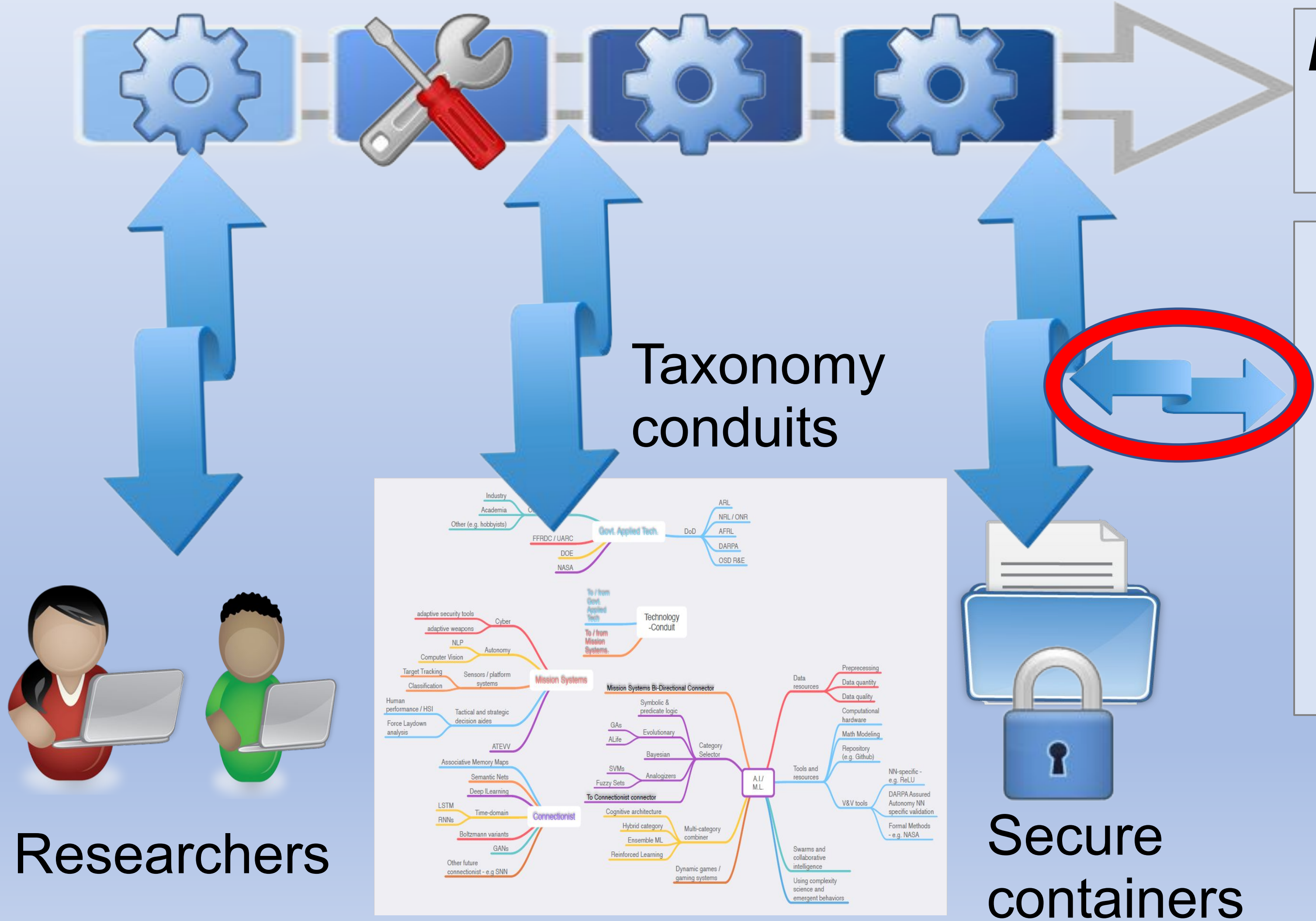
Even if the ETE architecture is incomplete, now is the time to design a "universal" production system designed for adaptation and validation. Questions to be asked:

1. Is the research current state of the art?
2. Who is doing various parts of this research?
3. How do we avoid the "valley of death" common to research transition?
4. Can information flow effectively to / from researchers and customers?
5. What conduits exist for resilient & consistent software to transition to customer use cases?

Joe Schaff, NAVAIR / NAWCAD Mission Systems

Pipeline Architecture & Taxonomy Connections:

***Enhanced* DevSecOps = researchers, tools, taxonomy conduits, secure containers.**



Prototype ready for customer

Customer Use Cases:

Connected via automated UML generated from selected software containers

Joe Schaff, NAVAIR / NAWCAD Mission Systems

Issues and What's Next?

<u>The "big picture" is currently incomplete:</u>

- Segments of the ETE architecture exist, satisfy some gaps.
- Other gaps exist: both known and unknown.
- Where does complexity provide advantages? Where are deterministic solutions better?
- Must work in a multi-domain battlespace – the two ends (swarm, components) are designed specifically for that.
- What organizations can address the "big picture"?
- Now at critical junction for MUmT and autonomy – incomplete/delayed response could put us too far behind adversaries to catch up.

Final Assessment

On-going work – things I am doing so far:

- Developed a class of algorithms that manage massive "smart" swarms:
  - Similar approach to ecosystems in nature, "stigmergic" communication.
  - Leverages "swarm intelligence" = **AI**, so that any entity "knows" where the others are positioned, as well as changes when broadcasted.
  - Needs ***only a few bytes of data*** to reorganize / know relative positioning of all battlespace entities.
  - Trivial math – e.g. raspberry Pi can calculate 10,000+ entities positions & dynamics in less than 100µsec.
- Developed the resilient meta-reasoning architecture for components:
  - Uses heterogeneous **AI / ML** algorithms in a complementary manner = weakness of one type of algorithm is covered by another, + helper functions for learning as needed. **Scales from *raspberry Pi* to largest available**.
  - AI algorithms are given free reign in a "sandboxed" environment to allow the full creativity or innovative results for most effective tactical decisions.
  - Meta-reasoner is the "rationalizer" or "adult supervision" that decides whether an algorithm has been deceived, choosing another algorithm's results if needed. Periodically, meta-reasoner learns and adapts.
- Ongoing collaboration with NASA LaRC Formal Methods laboratory.
- Ongoing collaboration with academia, DARPA Assured Autonomy, OFFSET programs.
- Tech & taxonomy architecture for transition.

BACKUP SLIDES

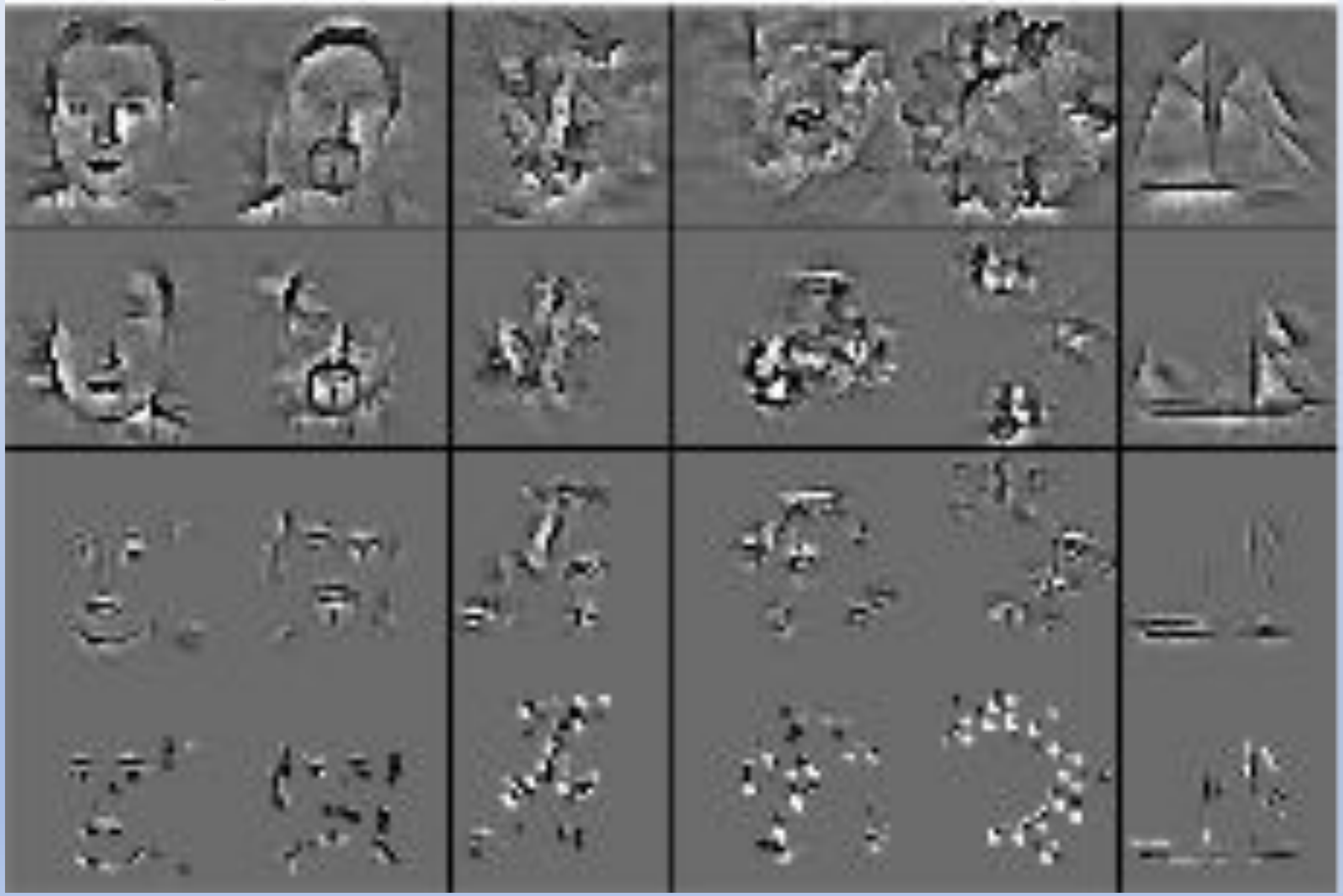
The Details…

Adversarial AI

Adversarial AI Malware

1. Extracted from Hu and Tan: "*Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN*"
   - Works even when attackers have no access to the architecture and weights of the neural network to be attacked.
2. Extracted from paper by UMD researchers: "*Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks* "
   - **Data poisoning = attack on machine learning (ML).**
   - Attacker adds examples to training set to manipulate the behavior of the model.
   - Targeted to control the behavior of the classifier on a *specific* test instance without degrading overall classifier performance.
   - Attacker adds a seemingly innocuous image (that is properly labeled) to a training set for face recognition, and control the identity of a chosen person.
   - ***Poisons*** could be entered into the training set simply by leaving them on the web and waiting for them to be scraped by a data collection bot.
3. **Images in nature can confound machines.**

Machine Deconstruction: Deconvolutional Network for Face Decomposition



- Top-down parts-based image decomposition with an adaptive deconvolutional network. Each column corresponds to a different input image under the same model.
- *low-level edges, mid-level edge junctions, high-level object parts and complete objects*

*{extracted from: Zeiler, Taylor, Fergus; "*Adaptive Deconvolutional Networks for Mid and High Level Feature Learning*"}*

Constructing a Deep Fake

Several methods to construct deep fakes – some use Generative Adversarial Networks (GANs), other methods for deconstruct / reconstruct facial features.

Cyber Resilience

Understanding Differences Between Cyber - {*Security*} and {*Resilience*}

Security:

1) Preserving data ”at rest” and in-transit.
2) Privacy = encryption, least-privilege access.
3) Securing system against external attack – hostile takeover, network-based attacks, etc.

Resilience:

1) More AI / ML based problems.
2) Resilient to deception / misclassification.
3) Resilient to noise added to data.
4) Recovery from exploitation of known weaknesses in classifiers.
5) Recovery from unanticipated attacks.

Joe Schaff, NAVAIR / NAWCAD Mission Systems

<u>Steps 1 and 2</u>: Cyber-secure Kernel, Linux Containers

1) **Use a microkernel OS =** ***Example:*** **Fuchsia (by Google – in development).**
   a) Based on a new microkernel called "Zircon" secure computing environment.
   b) Similar approach used by DARPA High Assurance Cyber Military Systems (HACMS) program.

2) **Use Linux Containers (e.g. "Docker")**
   a) Why?
      1. It "sandboxes" unstable or vulnerable, yet useful ML algorithms.
      2. Sandbox can re-instantiate the algorithm if it "crashes" due to malicious attack or instability.
      3. Allows full creativity or "emergent behaviors" of algorithms.
   b) Overhead and stability costs?
      a) Almost identical to bare metal or native ML application without sandboxing.
      b) If container crashes, then microkernel restarts container app with "sandboxed" algorithm.

Pipeline Architecture: R&D to customer Conduits

Pipeline Architecture: A Multi-pronged Approach

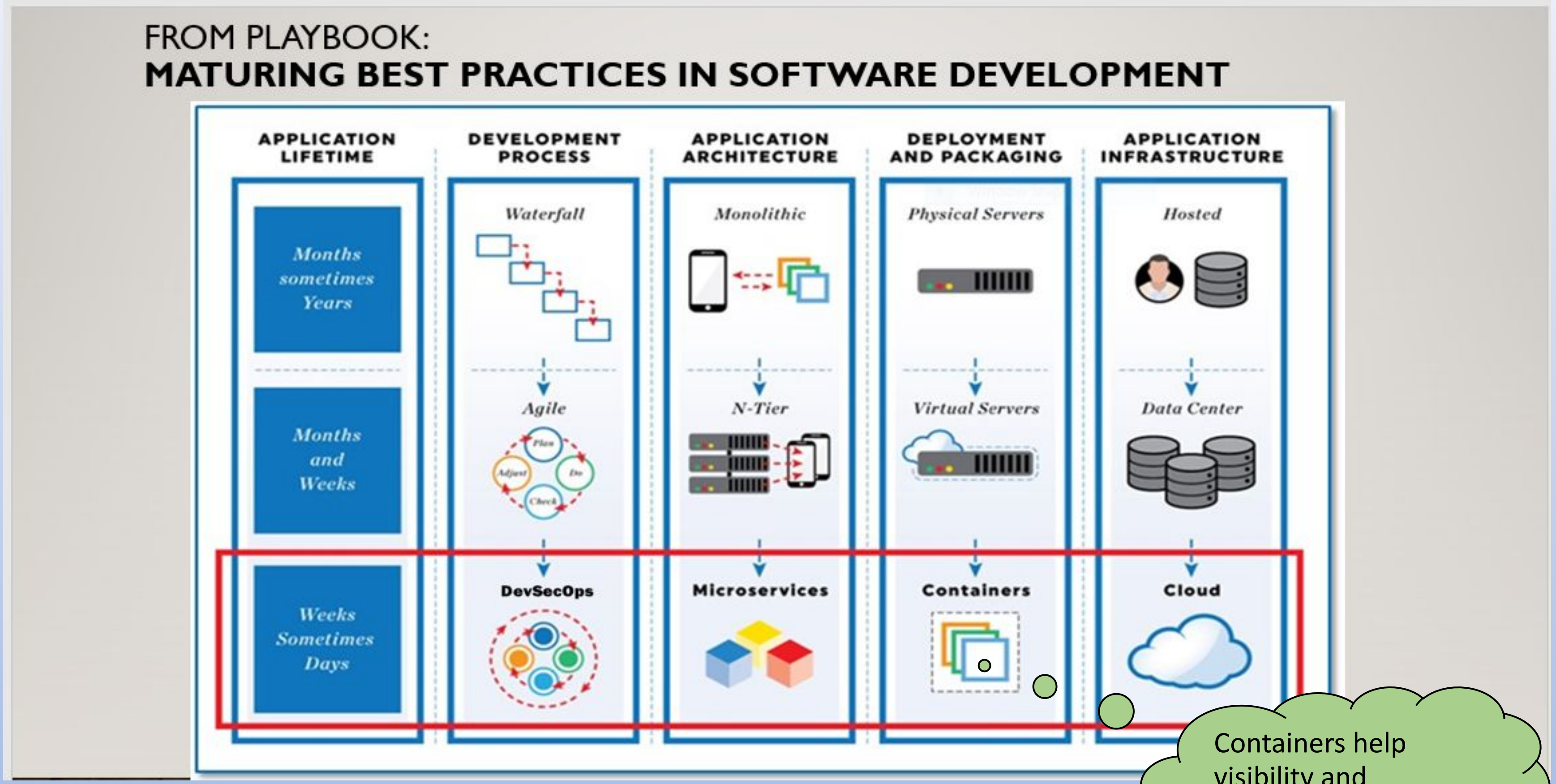
1. <u>**Foundation:**</u> create developer pipelines, i.e. - remove any burden of operations so that ***researchers concentrate on research***.

2. **Latest technology advances** from all available sources = ***follow the taxonomy tree***.

3. **Identify gaps** and unfulfilled needs = *where to invest in the research effort*.

4. **Map use cases** to UML / MBSE language abstraction of software, for transition pipeline.

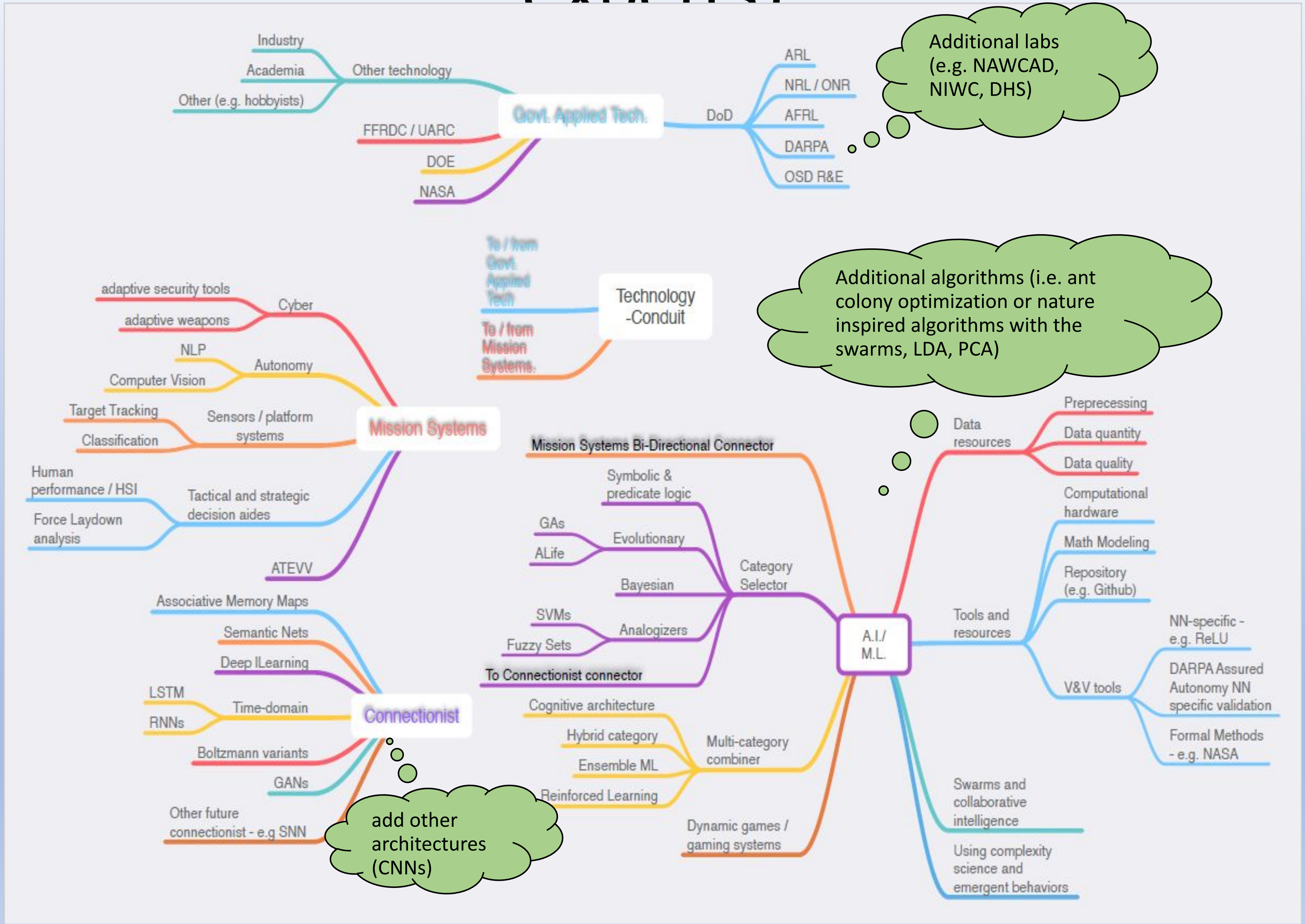
OSD DevSecOps & more?



Containers help visibility and sharability of products.

- Pipelines to / from developers.
- Hardened containers for algorithms or other software components.
- MilCloud based = latest research in AI/ML may be shared with other researchers.
- <u>***BUT…this pipeline is not enough. Need to insert taxonomy…***</u>

Taxonomy (with technology *conduits* to domain experts)

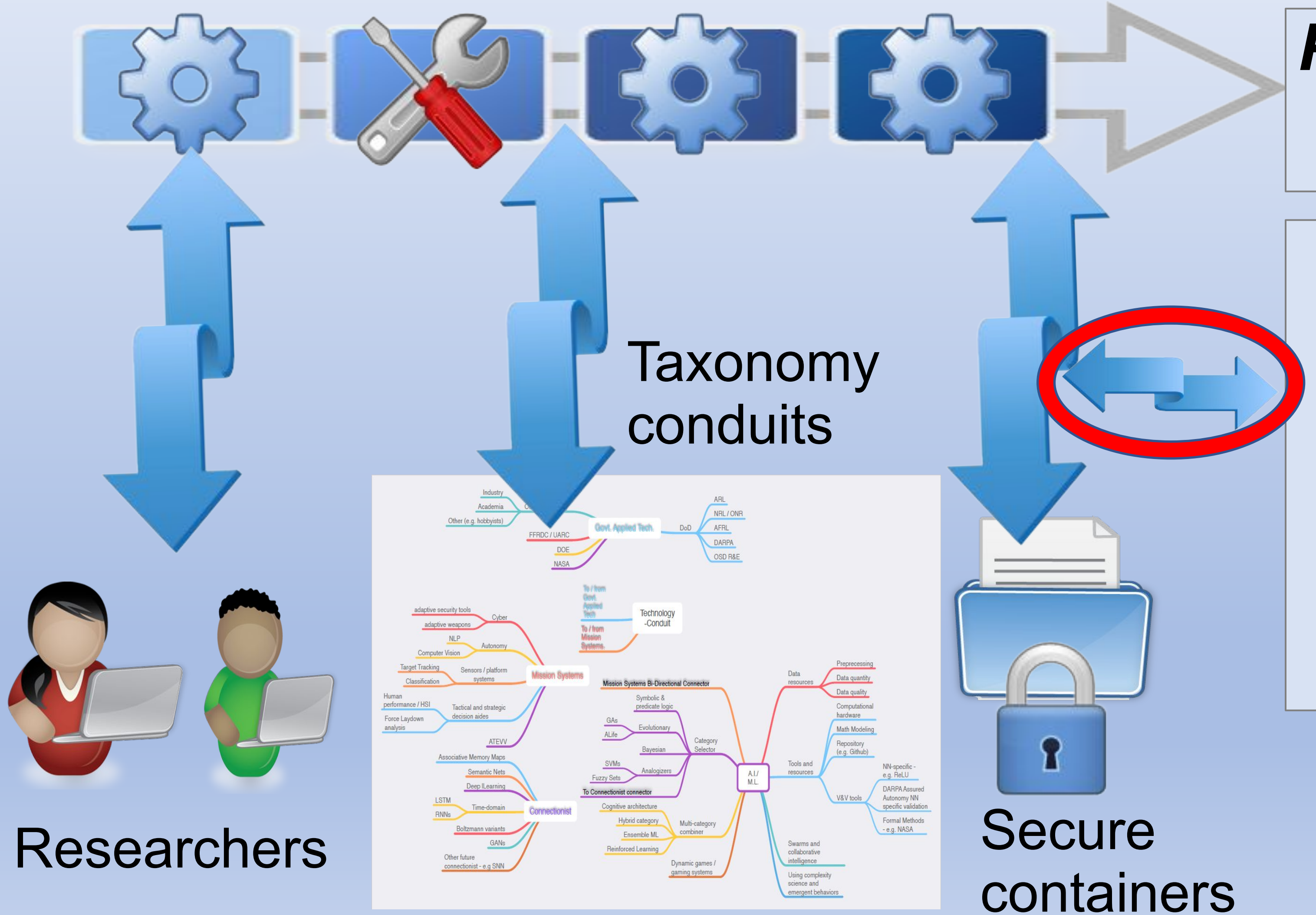


Joe Schaff, NAVAIR / NAWCAD Mission Systems

DISTRIBUTION STATEMENT A

Pipeline Architecture & Taxonomy Connections:

***Enhanced* DevSecOps = researchers, tools, taxonomy conduits, secure containers.**



Prototype ready for customer

Customer Use Cases:

Connected via automated UML generated from selected software containers

Joe Schaff, NAVAIR / NAWCAD Mission Systems

Human-Robot Interaction Course
(I designed & teach this @ U. of Maryland)

Course Outline1

- **<u>Course will cover topics</u>** as diverse as the technology for biologically inspired robots, cognitive robotics, cultural, social and legal aspects of robotics, data mining, examples of human systems interfacing, machine learning principles and their limitations with respect to AI.
- **<u>Your objective as a student</u>** will be to integrate this interdisciplinary knowledge and perform ***out of the box*** thinking, demonstrating this in a ***term project***.
- We're going to look at the ideas like robot emotion, and collaborative robots that can form limited social interactions.
- **<u>You will design a robot</u>** that can ***implicitly determine*** the action it needs to take without explicit commands given to it, by observing its interaction with people.

Course Outline 2

- **<u>The term project:</u>** Think of creating a Kickstarter where you will be building the next generation of cognitive human-behaving robots.
- You need to show your product as something investors would buy into.
- I will provide course material and extensive reference sources for both hardware and software to design these robots.
- These robots could realistically be built with hardware and software for as little as $2000.
- The Kickstarter is only a goal to shoot for, and if you indeed want to create an actual one after the course is over, you are encouraged to do so either alone or in collaboration with others in your class.
- Unlike an actual Kickstarter, there's no penalty for not being sponsored - if you try and think out of the box, and apply whatever knowledge you're capable of finding as well as what I will provide, **<u>you will succeed.</u>**