



# Developing AI/ML Components (and their Standards) for Civil Aviation : Challenges and Barriers

Natasha Neogi

NASA Langley Research Center

AVIATE Seminar

January 24, 2025

# Outline

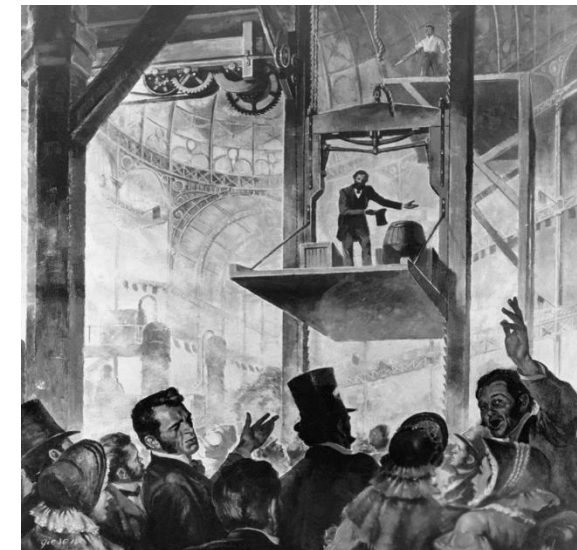
- Motivation and Definitions: Why are we here?
- Key Question: What constitutes sufficient evidence that an ML component meets its requirements?
- Questions to answer and a way forward...



# Motivation and Definitions: Why are we here?



Elevators carry two billion passengers a day over hundreds of millions of vertical miles in over 200 nations.



Courtesy of Unknown author - Copie de gravure ancienne, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=30135037>

# Problem and Goal

## Problem

- The inability to establish appropriate assurance for AI/ML components leaves us unable to effectively manage their risks and benefits.
  - Drives cost of development uneconomically high
  - Delays adoption of AI/ML at scale in safety critical systems
  - Results in unknown and unmanageable risks

## Goal

- Discover and define what constitutes sufficient evidence to substantiate a safety claim related to an AI/ML component performing a safety-critical function.

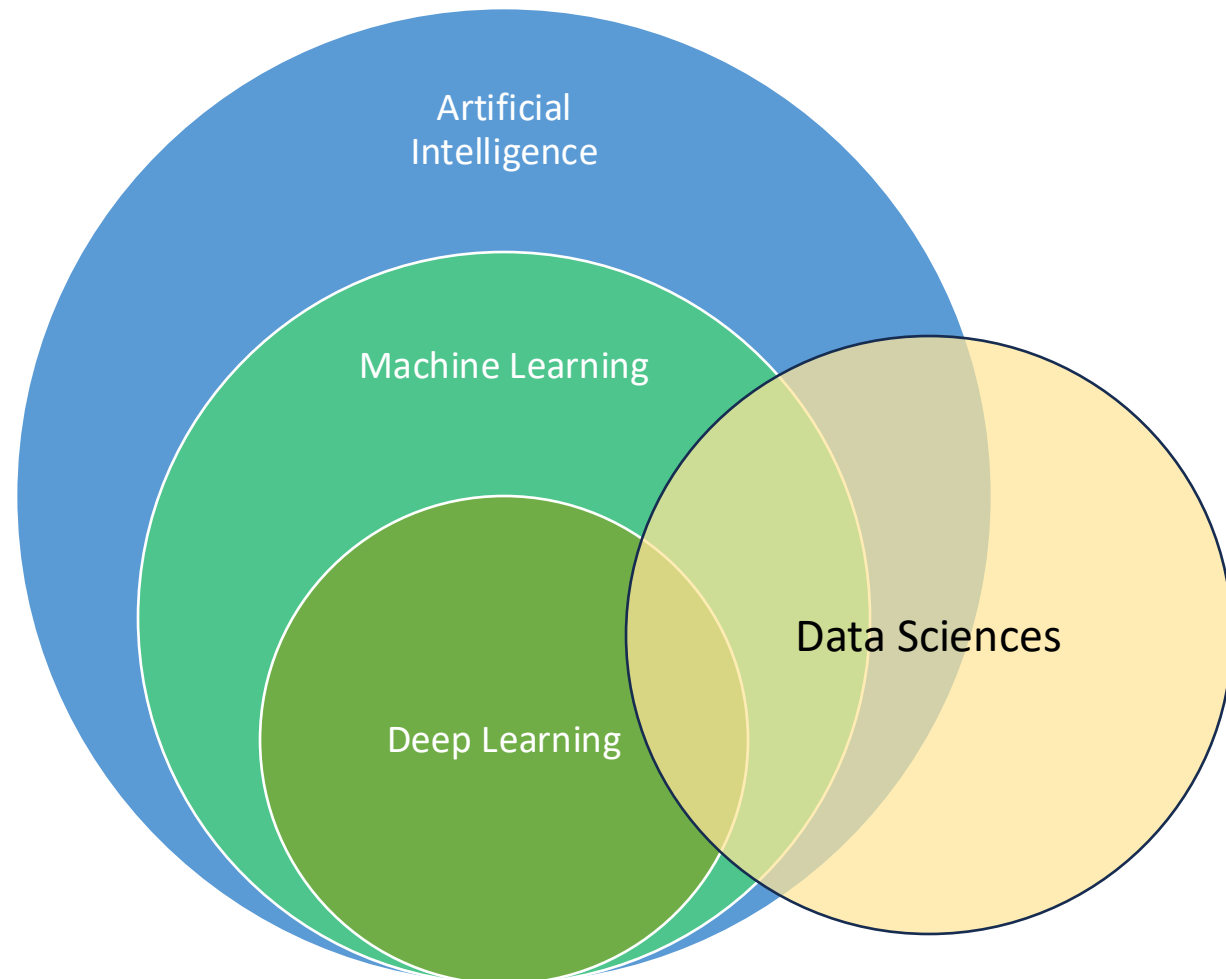


# Towards standards for AI/ML...

- Standards require a stakeholder consensus on a driving need, commitment to support development, and subsequent application.
  - Broad sector of stakeholders should be involved, or uptake will suffer
  - SME contributions from all relevant or impacted stakeholders (e.g. aircraft OEMs, avionics OEMs, airline operations, pilots, controllers, researchers, regulators, etc.)
- Standards development activities require a keen and deep understanding of the problem being solved and the technologies deployed in its reference implementation.
  - Understand mechanisms and limits of fundamental, underlying science of implementation and verification technologies
- NASA's Role is to:
  - Provide SME(s) to help create the standard;
  - Provide relevant findings from NASA R&D activities;
  - Learn of gaps or challenges that need to be addressed, then start up (or change) R&D activities to help fill these gaps or address the challenges.

# Definitions: AI and ML

- AI: “AI is the science of making machines do things that would require intelligence if done by men.”
  - Marvin Minsky, 1956
- ML: Use of statistical techniques to analyze data and create algorithms that can generalize to unseen data without explicit programming
  - <https://gradml.mit.edu/intro/>





# When I say AI, I mean...

## Machine Learning Systems

- Reinforcement Learning, Supervised Learning, Unsupervised Learning, Generative Systems...

## Rule Based Learning Systems

- Production Systems, Expert Systems, Fuzzy Logic,...

## Search and Optimization Techniques

- Uninformed Search, Informed Search, Parallel Search,...

## Decision Making under Uncertainty

- Bayesian Inference, Parameter Learning, Structured Learning,...

## Evolutionary Strategy

- Evolutionary Algorithms, Swarm Based Algorithms,...





# When I say ML I mean...

Supervised Learning

- Regression, Classification, ...

Unsupervised Learning

- Dimensionality Reduction, Clustering, ...

Weakly Supervised Learning

- Transductive Learning, Inductive Learning, ...

Reinforcement Learning

- Q-Learning, Policy Gradient, Deep Q networks,...

Generative "AI"

- Variational AutoEncoders, Generative Adversarial Networks, Transformer Networks,...



# But wait! ML is already here...

- Use of optimization tools during design of Boeing 787
  - Wing-Body Design
  - Composite Material Design
- Initial efforts at NASA to deploy ML techniques in aviation contexts
  - Using Large Language Models to look for positive contributions to safety in ASRS reports
  - Anomaly Detection/Vulnerability Discovery
- and others...



Courtesy of Timo Breidenstein –  
[http://www.airliners.net/photo/United-Airlines/ Boeing-787-822-Dreamliner/2142634/L/](http://www.airliners.net/photo/United-Airlines/Boeing-787-822-Dreamliner/2142634/L/), GFDL 1.2,  
<https://commons.wikimedia.org/w/index.php?curid=20544763>

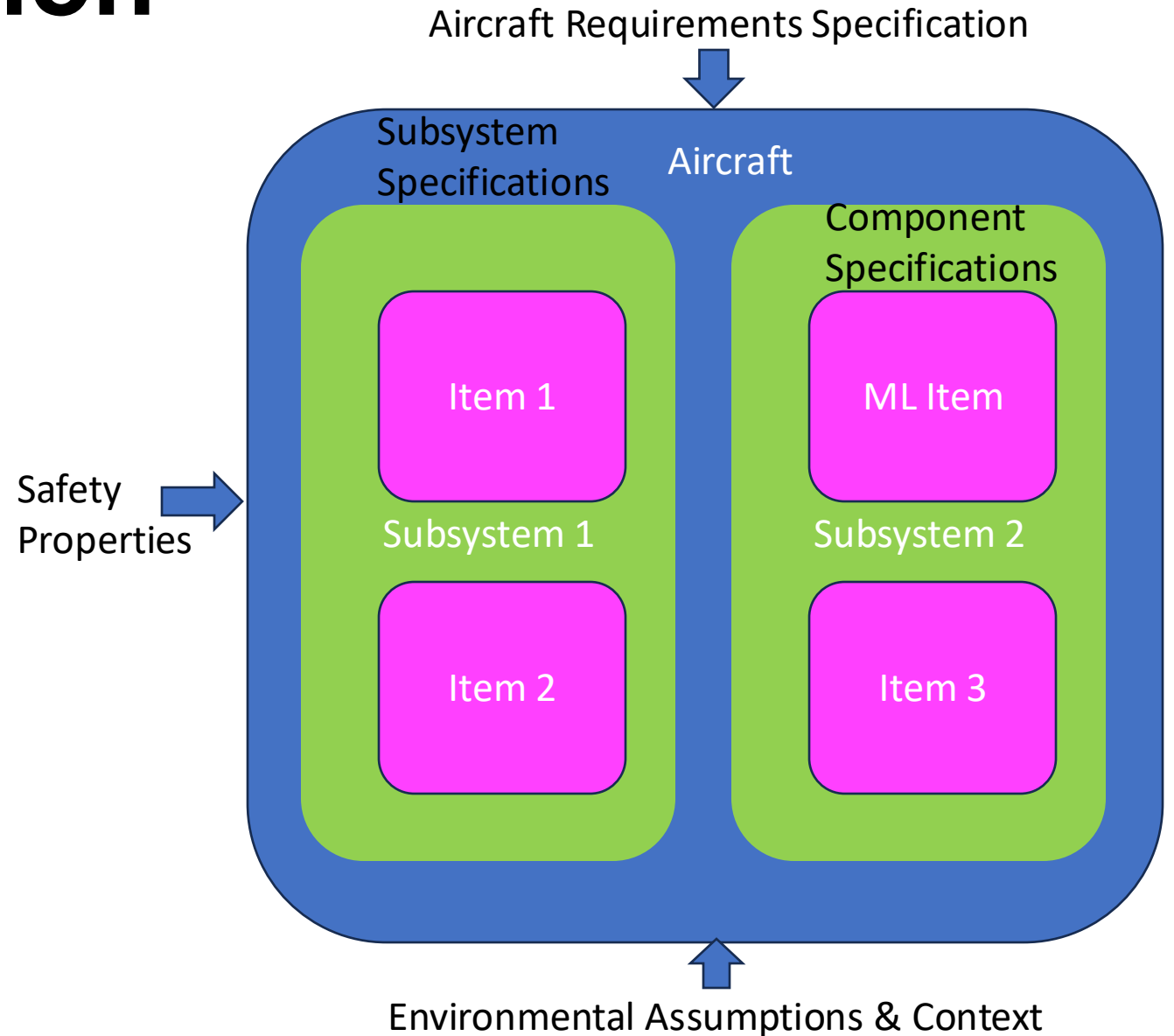


Key Question:

What constitutes sufficient evidence that an ML component meets its requirements?

# Aircraft Certification

Primary aim of aircraft certification (Part 21 etc.) is to provide **assurance of safety** by: (1) assuring that items perform their intended (safe) functions under **any foreseeable operating condition**, and (2) assuring that unintended functions are improbable.



# The three “E’s”

- **Explicit Claims**
  - Required emergent properties must follow from the combination of the properties of the system component (that is, ML component implementation) and the domain assumptions (context); environmental assumptions (including interfaces); and constraints.
  - They should indicate explicitly the level of assurance claimed.
- **Evidence**
  - Concrete evidence is usually a combination of testing, analysis (including modelling and simulation), and appeals to process.
    - e.g., software deployed in the field is the same as software under test/analysis
- **Expertise**
  - Developers should be familiar with best practices and deviate from them only when needed.
  - Experts can wisely tailor their approach to assuring novel elements with respect to methods, languages, tools, and processes.

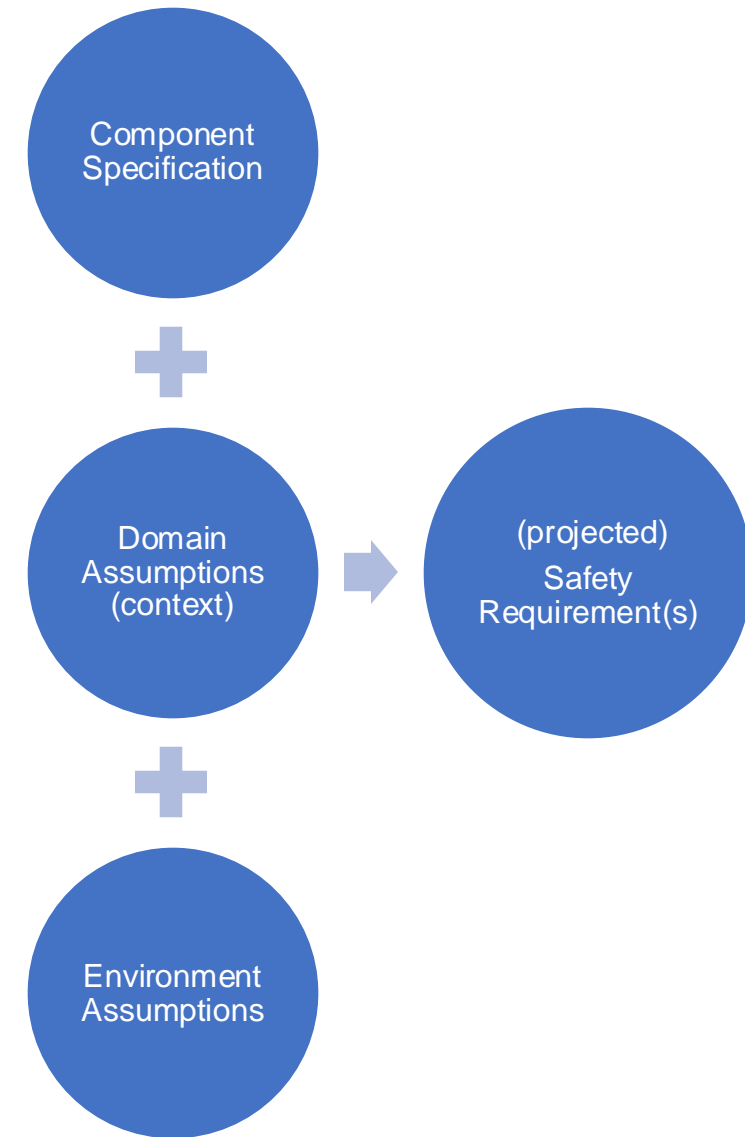


# Explicit Claims: ML as a component of a system

- Safety is not an intrinsic property of an ML component.
  - An ML component may be safe in the context of one system but not in the context of another.
- The specification of an ML component characterizes the behavior of the ML software at its interface with other system components and the environment.
  - It is important to distinguish this specification from the desired emergent (safety) properties of the system in the physical world.
- If a ML component only meets its assurance criteria if humans interacting with the system behave in a certain way, then this becomes an assumption on the environment of the component (or a constraint of system components) that must be evaluated for validity.

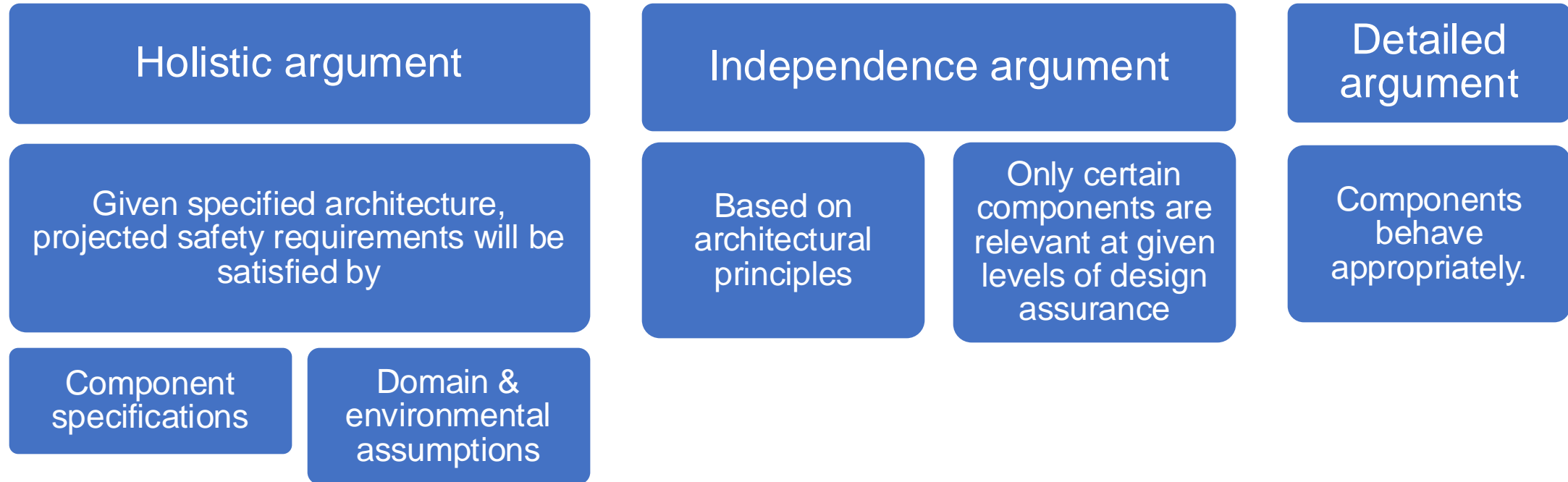
# Explicit Claims: The role of assumptions

- Domain Assumptions (context): An explicit statement of domain assumptions (context) of the ML component is required to evaluate any safety argument associated with that component.
  - This requires an argument that the specification of the ML component and the domain assumptions together imply the projection of the desired emergent property on the component.
  - Must perform all three activities: (1) check the ML component, (2) check the domain assumptions, and (3) check that they have the correct combined effect.
- Environmental Assumptions: An explicit statement of environmental assumptions (often physical parameters outside the scope of design authority) is needed to evaluate any safety argument associated with a component.
  - These assumptions must be validated at design time and during operations.



# Explicit Claims: The role of architecture

- The case that a system with an ML component with architectural mitigations satisfies a safety claim (or requirement) may proceed as follows.



- Note that restrictions related to architectures and relief from required design assurance levels specified in current standards must be considered in such an argument.



# Evidence: Why traditional techniques do not translate for ML

	Traditional (Physical) Components	ML (Software) Components
<b>Criteria</b>	Criteria are simple (e.g., failure/breakage rate etc.) for the component as a whole.	Complexity of ML and its interdependence on its domain and environment make it difficult to have explicit and precise articulation of meaningful criteria that can be measured.
<b>Feasibility of Testing</b>	For physical artifacts, limited testing provides compelling evidence of quality, with the continuity of physical phenomena allowing widespread inferences to be drawn from only a few sample points.	Limited testing of ML cannot provide compelling evidence of behavior under all conditions.
<b>Process &amp; Product Correlation</b>	Underlying principle of statistical quality control is that sampling the product coming out of a process gives a measure of the quality of the process itself, which in turn will determine the quality of items that are not sampled.	More rigorous ML design processes will likely lead to better quality ML components. However, this correlation is not sufficient as the sole provider of evidence, as correlation does not imply causation.

# Evidence: Testing, simulation and analysis, and formal methods

- Testing for ML components is indispensable.
  - However, testing alone is insufficient, as it is unclear what coverage means in terms of ML components.
- Simulation and analysis can provide needed checks for ML components.

Validation of environmental assumptions, interface assumptions, and constraints

Feasibility or satisfiability analysis of temporal behaviors

Verification of code implementation against component specifications

Checking that components in aggregate achieve appropriate system-level effects

- However, simulation and analysis is insufficient due to model inaccuracy, incorrect assumptions (e.g., environmental, operator response, execution platform), etc.
- Formal methods can provide guarantees for ML components.
  - Formal methods can provide formal proofs of correctness.
  - Formal methods techniques often lack scalability.

# Expertise: Transparency and credibility of claims

- To establish that a system containing an ML component is safe will involve inspection and analysis of the safety claim and the evidence offered in its support.
  - Assurance of ML components requires explicit safety claims (or assurance requirements), evidence for those claims, and a rigorous argument that demonstrates that the evidence is sufficient to establish the validity of the claims (or satisfaction of the requirements).
- Evaluator should be able to calibrate not only the technical claims and evidence but also the organization that produced them, because the integrity of the evidence chain is vital and cannot easily be assessed without supporting data.

Qualifications of the personnel involved in the development of the ML component

Track record of the organization in providing ML components

Process by which the ML component was developed

Process by which data used to train/test the ML component is collected/curated/maintained etc.

# Expertise: Managing Complexity

- Only implement ML when necessary.
  - The key to achieving requisite assurance at reasonable cost is simplicity, including simplicity of critical functions and simplicity in system interactions.
- Use architectural means to mitigate complexity caused by ML when possible.
  - Establish independence so system level properties are guaranteed by individual components which preserve the emergent property despite failures in the rest of the system.
- Use rigorous processes to develop ML.
  - Each step in developing the ML software needs to preserve the chain of evidence on which will be based the argument that the resulting ML component meets its requirements (and the overall system is safe).

Questions to answer and a way  
forward...

# Questions to answer (I)

How do we know when an ML component's behavior meets its requirements?

- Sufficient representation and size of training dataset, accuracy vs. generalizability, what constitutes an actionable specification, etc.

What are the limits of current processes and metrics currently used in developing and evaluating both traditional and ML systems?

- How do you use testing (i.e., creating logical based oracles, etc.), simulation (i.e., model validity), (formal) analysis (i.e., scalability), runtime verification frameworks, etc. in assurance?

What are the set of characteristics and parameters of an ML system that allows you to bound its behavior (e.g., capabilities, limitations, etc.)?

- Data and information quality, architecture, associated metrics, etc.

What is the minimum set of information required to reconstruct and audit a ML implementation in the case of an accident?

- State, Environmental, and Input Information, Decision Making Logic, Configuration Management, Version Control, etc.
- How do you create an encoding scheme that would reduce the volume of state information into a tractable, compact form?



# Questions to answer (II)

How should information assurance be handled for ML components in order to yield (composable) safe systems?

- Data fusion; information synthesis; data collection, curation, and assurance; data poisoning; etc.

How can change be managed in ML systems in order to preserve assurance?

- Configuration management, version control, database management, etc.
- Full recertification, continuous authorization to operate, etc.

Can an actionable specification for a function be extracted from a dataset?

- Functional requirements, Safety requirements, Environmental assumptions, Domain specific constraints, etc.

When can ML be used in the design, development and/or accident/incident analysis process?

- Tool qualification (DO-330), ASRS database querying for research, prognostics/diagnostics, scheduling, maintenance, etc.



# Going Forward: Deploying ML in aviation systems

- Start with ML in design/analysis/maintenance (offline system, offline learning)
  - Start with simple, well-defined, non-safety critical applications
    - Recognizing normal and anomalous patterns in large datasets for research purposes (collaborative),
    - Querying large databases for research purposes (ASRS/ASAP), etc.
- Progress to ML in embedded flight/operational systems (online system, offline learning)
  - Start with functions which have
    - Clearly defined requirements,
    - Means of checking the answer/output, and
    - Means of intervention and mitigation of incorrect answers/outputs.



# Going Forward: Standards development for ML in aviation systems

- Identify current and ongoing standards efforts that may be applicable (e.g., DO-330, etc.) to ML components.
  - Leverage other standards bodies when appropriate to avoid duplicative efforts.
  - Standards efforts should be targeted at areas in which gaps are found.
- A measured approach to standards development for ML applications should target those functions for which there are actionable specifications and traditional implementation and assurance techniques.
  - Standardize criteria for what constitutes sufficient evidence for ML safety.



# Takeaways

- Deployment and standardization efforts should proceed methodically and with a justifiable basis, thereby enabling safe adoption of ML applications in aviation.
- Premature efforts to (deploy and) standardize may damage paths to transition for ML technologies, engender technical debt, or set back the entire aviation industry.



# Questions?

[natasha.a.neogi@nasa.gov](mailto:natasha.a.neogi@nasa.gov)

# Towards standards for AI/ML...

- Standards require a stakeholder consensus on a driving need, commitment to support development, and subsequent application.
  - Broad sector of stakeholders should be involved, or uptake will suffer
  - SME contributions from all relevant or impacted stakeholders (e.g. aircraft OEMs, avionics OEMs, airline operations, pilots, controllers, researchers, regulators, etc.)
- Standards development activities require a keen and deep understanding of the problem being solved and the technologies deployed in its reference implementation.
  - Understand mechanisms and limits of fundamental, underlying science of implementation and verification technologies
- NASA's Role is to:
  - Provide SME(s) to help create the standard;
  - Provide relevant findings from NASA R&D activities;
  - Learn of gaps or challenges that need to be addressed, then start up (or change) R&D activities to help fill these gaps or address the challenges.

# Questions to answer (III)

What are key domain specific considerations that may dominate the safety of ML implementations and how will we address them?

- Lack of safe default mode/state, inability of pilot to intervene, etc.

What is the current human contribution to safety in the function being replaced by an ML/AI implementation (i.e., full extent of the capabilities and limitations of the human role)?

- Consider critical information dependencies across tasks executed collaboratively by diverse agents, etc.

Can the open world problem be solved (and standardized) without humans to handle edge cases while maintaining the current level of NAS safety?

- Handling epistemic uncertainty, applicability of real-world data across different environmental assumptions, etc.

What are the characteristics of a function that could help in assuring a ML implementation?

- Clear (and testable) set of requirements, outputs easily checked for correctness, corrective action can easily be taken, etc.