Assurance Cases as Knowledge Graphs

SOFTWARE CERTIFICATION CONSORTIUM MEETING

Annapolis, MD, May 9th, 2024

Mauricio Castillo-Effen, Ph.D.

Overview

A. Background

B. From Toulmin-inspired to “Digital” Assurance Cases

C. CertGATE

D. Knowledge Graphs

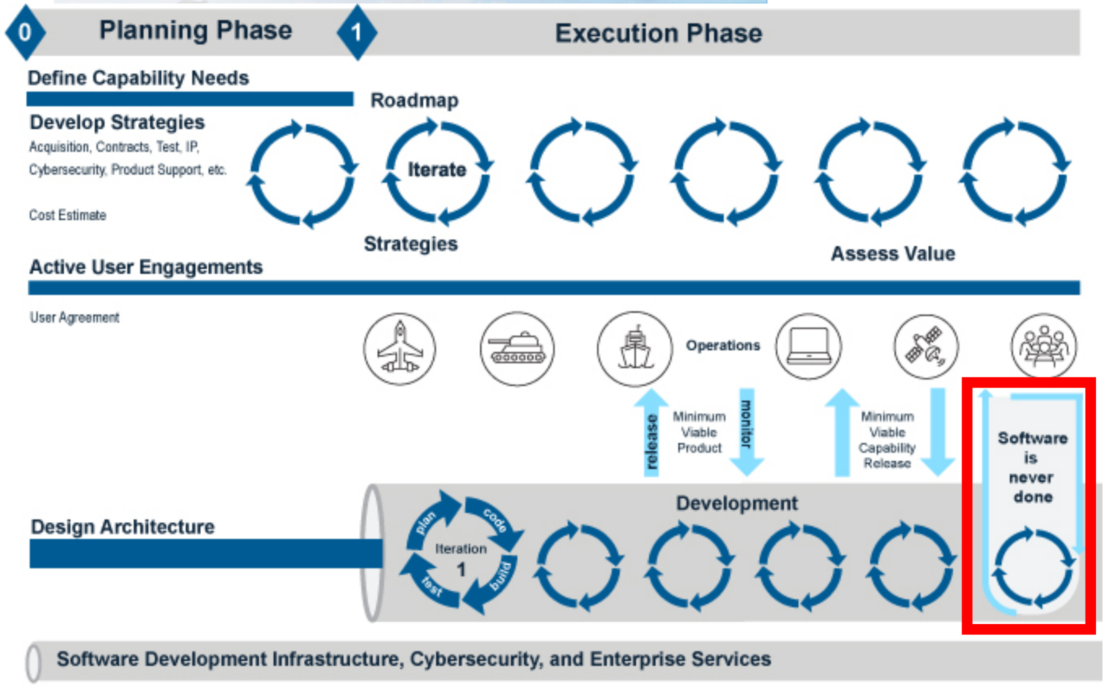
DISCLAIMER
The views and opinions presented in this presentation are solely the author’s, and they do not represent the official policy or position of the Lockheed Martin Corporation or the Lockheed Martin Advanced Technology Laboratories.

A. Background

Acquisition Demands

- Capabilities are software/AI-enabled
- Capabilities will be assessed and assembled **just-in-time** and **at the edge** to address specific mission needs
- Capabilities are added/composed and weaknesses addressed continually (“**software is never done**”) based on dynamic needs while **maximizing reuse**!
- Capabilities may be obtained from **multiple vendors and providers**
- Decisionmakers need to **assess risk-benefit tradeoffs** based in part on the information obtained from assurance activities
- Transition from traditional waterfall V-model systems engineering to Dev*Ops transforms the view of **assurance from compliance- to value-driven**
- This assurance **“ConOps” may apply to dynamic aspects of other high consequence systems** that may contain less dynamic components

LM
Project Hydra



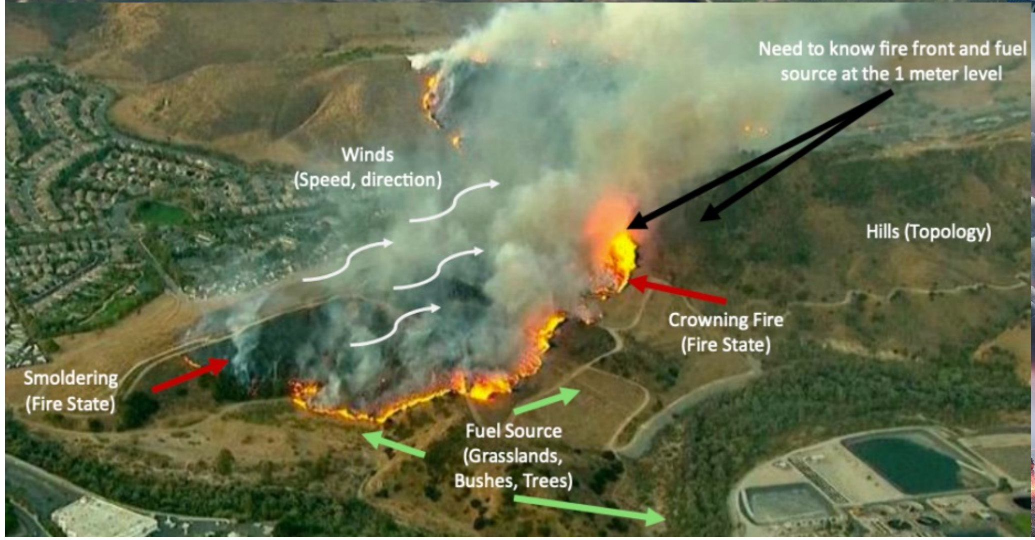
From DoD instruction 5000.87: OPERATION OF THE SOFTWARE ACQUISITION PATHWAY

New acquisition regimes are characterized by **complexity** and **agility**

LOCKHEED MARTIN

Autonomous Systems for Firefighting

This image was generated with the assistance of AI.



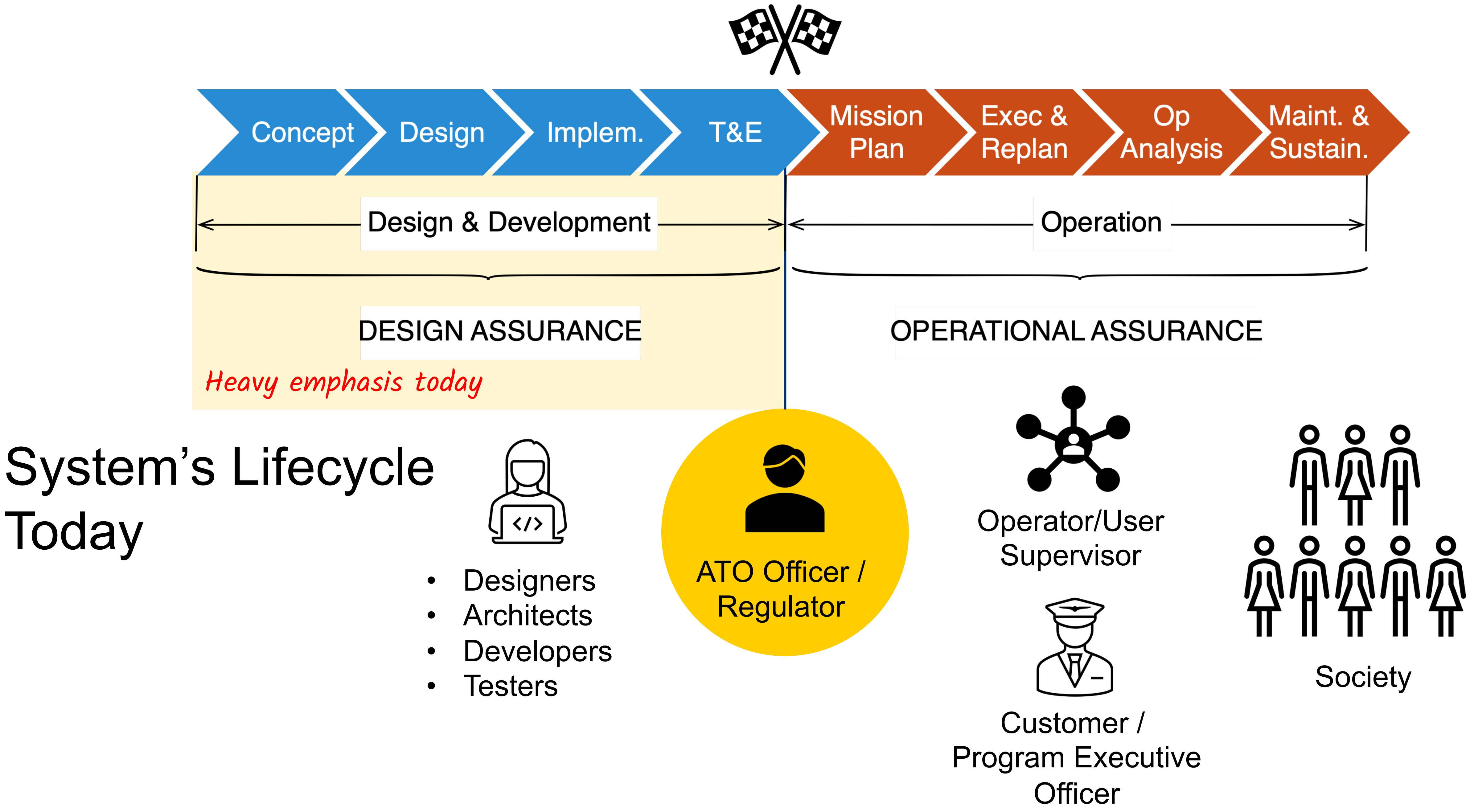
This image was generated with the assistance of AI.

https://blogs.nvidia.com/blog/2021/11/09/lockheed-martin-wildfires-ai/

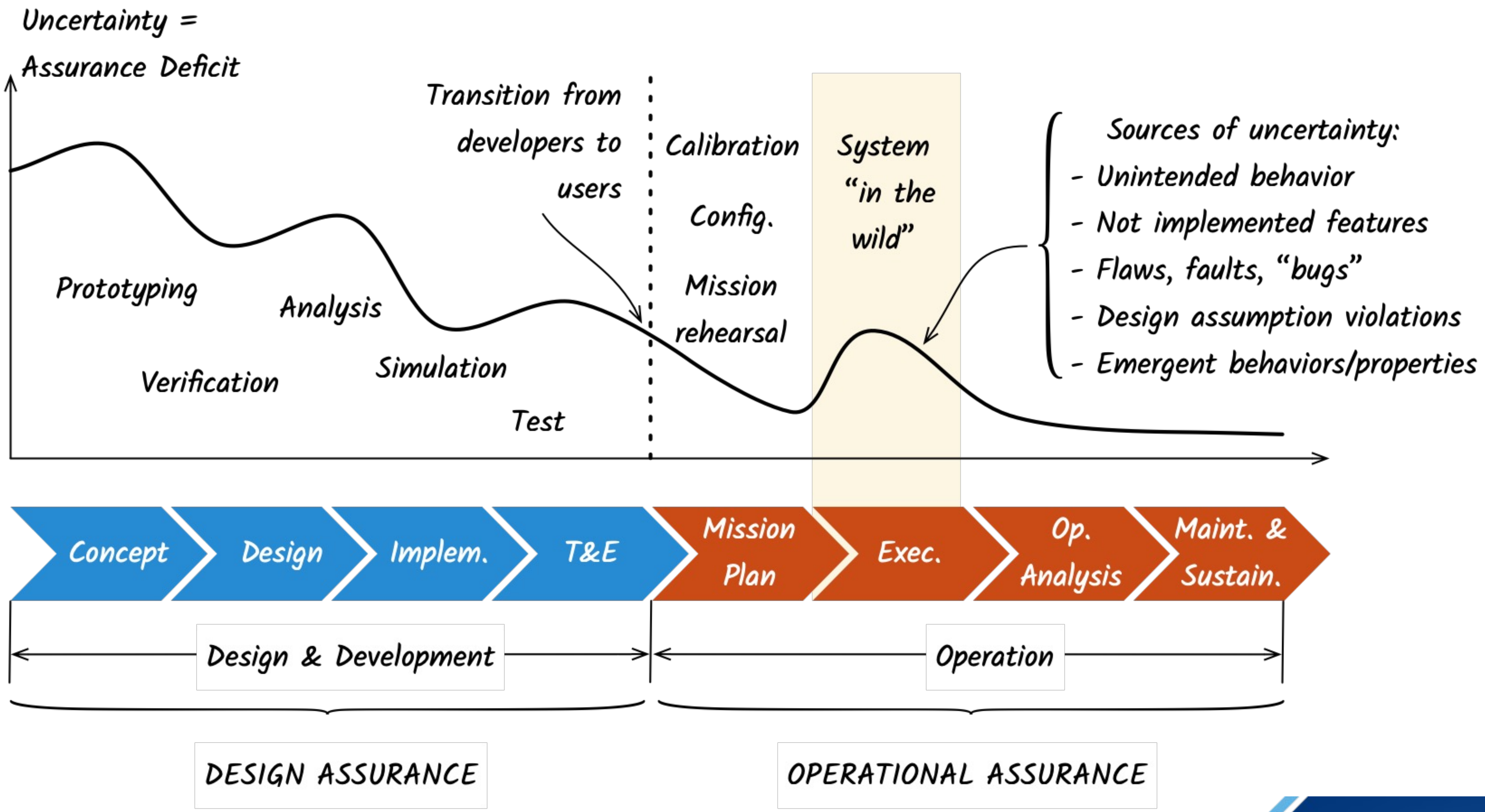
Exemplary Assurance Challenges

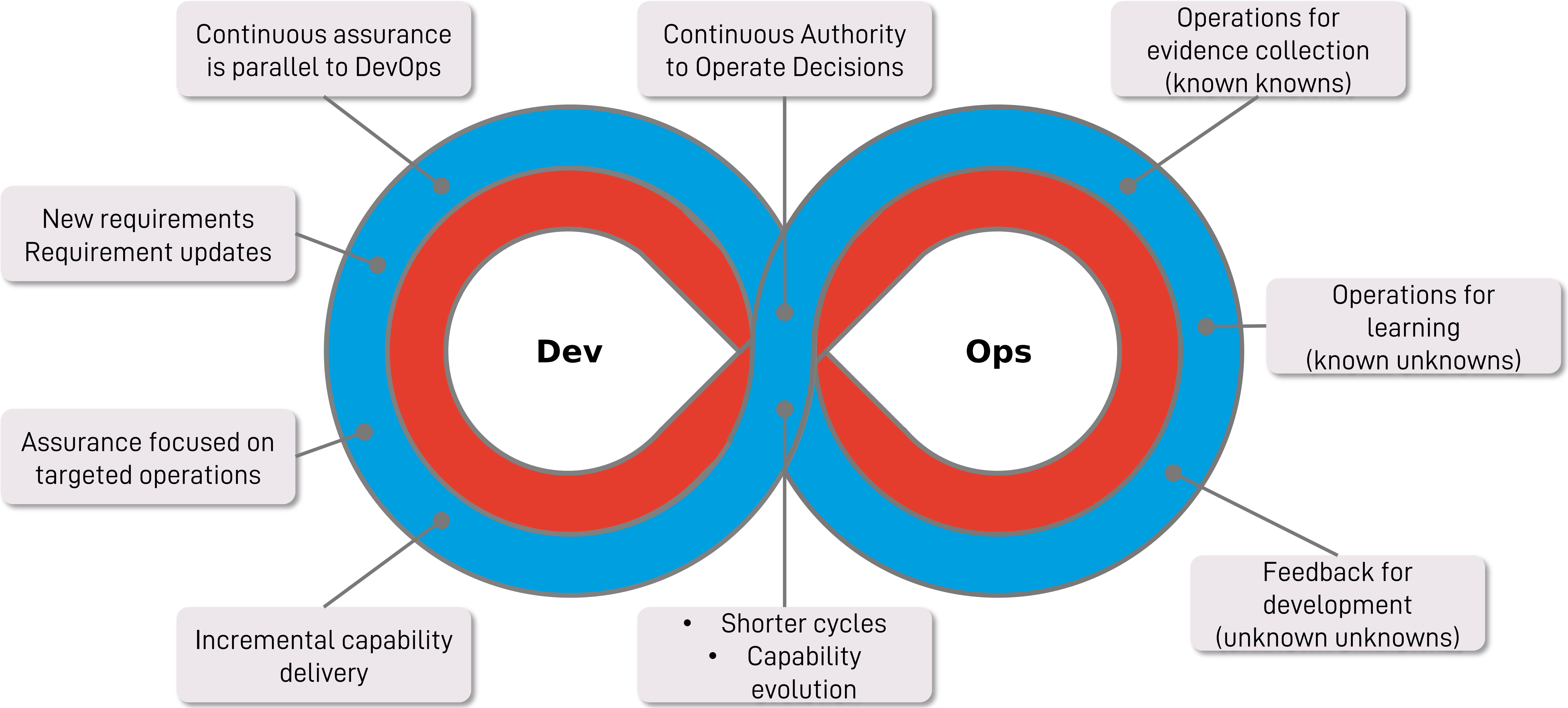
- Specifying and verifying correct and effective crewed-uncrewed heterogeneous team behavior in the presence of unique and rapidly evolving wildfire conditions
- Ensuring combined capabilities and resources (sensing, endurance, agility, payload, reliability, accuracy, etc.) applied in the firefighting mission are adequate and sufficient
- Maximize reuse
- Minimize unintended consequences
- Ensure the system is governable by a commander
- Ensure effective crewed-uncrewed collaboration
- Cyber resiliency

LOCKHEED MARTIN

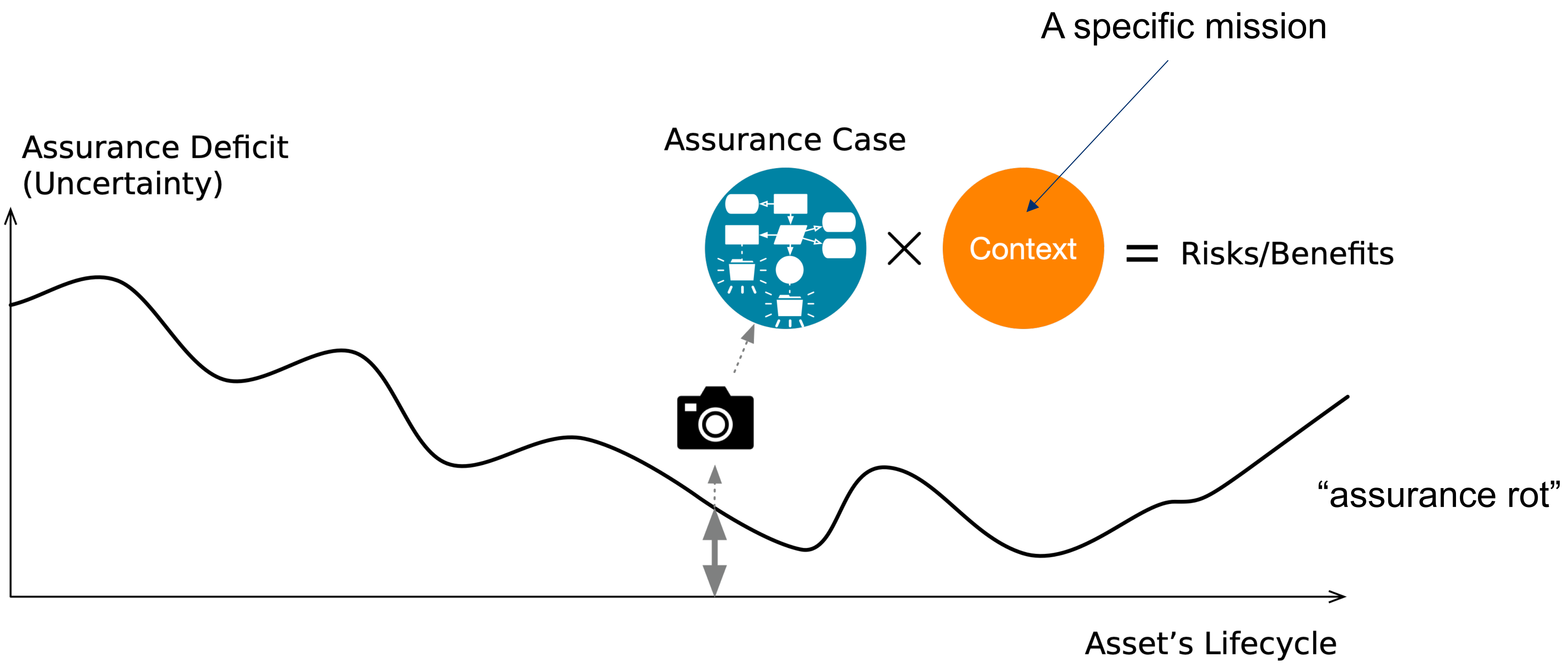

Concept
Design
Implem.
T&E
Mission Plan
Exec & Replan
Op Analysis
Maint. & Sustain.
Design & Development
Operation
DESIGN ASSURANCE
OPERATIONAL ASSURANCE
Heavy emphasis today
System's Lifecycle Today
Designers
Architects
Developers
Testers
ATO Officer / Regulator
Operator/User Supervisor
Customer / Program Executive Officer
Society

Assurance as Uncertainty Reduction




Continuous assurance is parallel to DevOps
Continuous Authority to Operate Decisions
Operations for evidence collection (known knowns)
New requirements
Requirement updates
Dev
Ops
Operations for learning (known unknowns)
Assurance focused on targeted operations
Feedback for development (unknown unknowns)
Incremental capability delivery
• Shorter cycles
• Capability evolution

Vision: Continuous Assurance Evaluation



This is what we want but can't do with assurance cases today

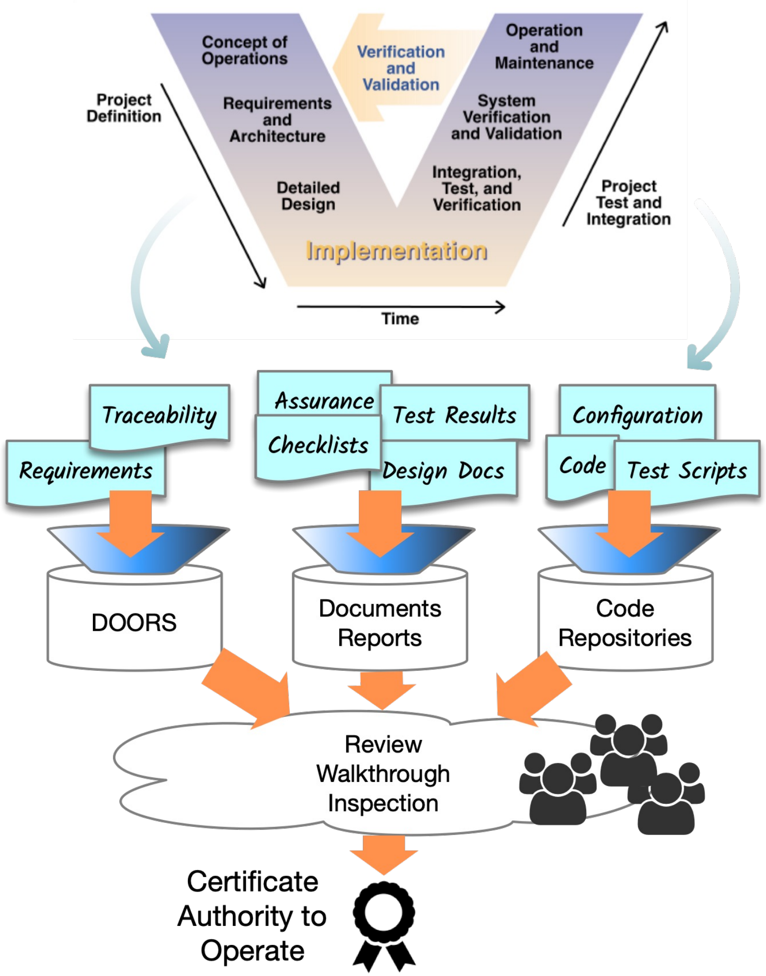
From Toulmin-inspired to Digital ACs

TODAY

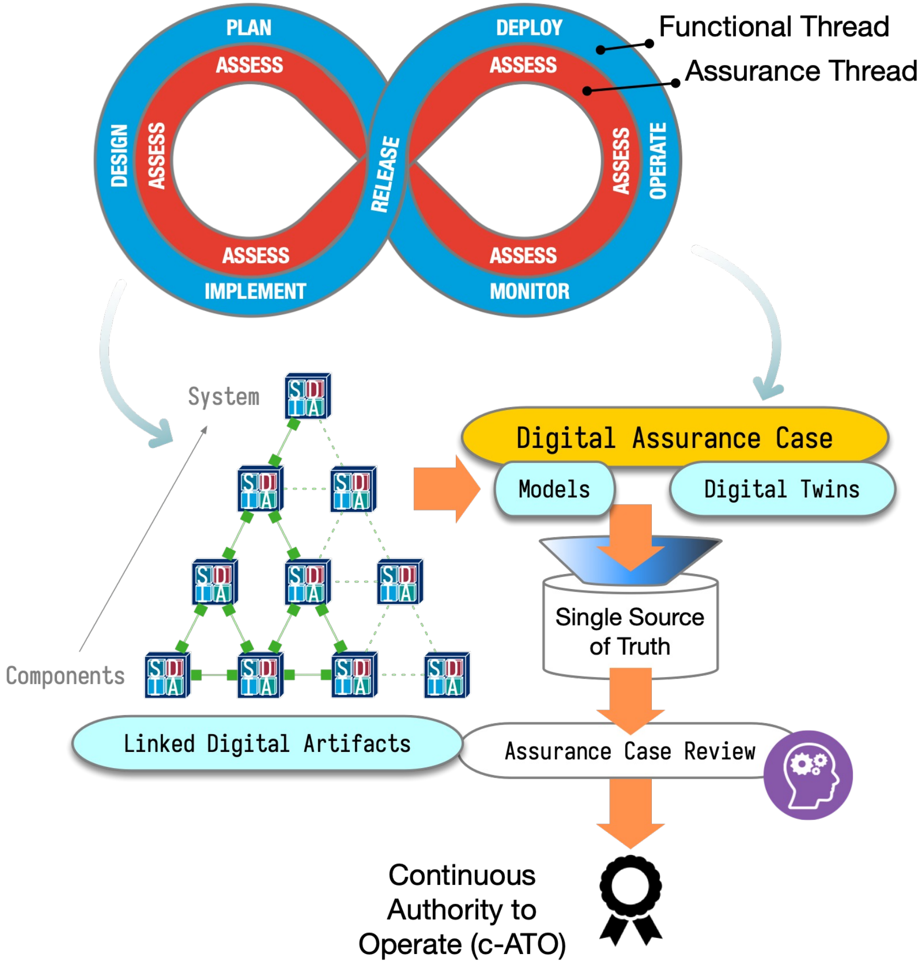
Methodology

Artifacts

Assessment



VISION



Digital Assurance Cases are Aligned with **Digital** Engineering Methodology, Artifacts, and Agility

LOCKHEED MARTIN

From Toulmin to Digital Assurance Cases

Digital Assurance Cases	Toulmin-inspired Assurance Cases
Concern Separation: Four separate and potentially distributed processes : i) Evidence Generation; ii) Assurance Case Construction; iii) Evaluation; iv) Presentation	ii)-iv) are intermingled and disjoint from i)
Inductive arguments: Suggest or hypothesize claim satisfaction. Assurance case evaluation leads to informed benefit/risk evaluation	Strive for deductive validity
Efficient Computational Constructs: focus efficient algorithmic storage and manipulation (querying, transformation) of evidence, arguments, and other metadata	Graphical/textual interactive editing (GSN, ACE, FAN, etc.)
Semantics: Use of controlled vocabularies (knowledge graphs), algorithmic semantic reasoning	Mostly, natural language for human readability

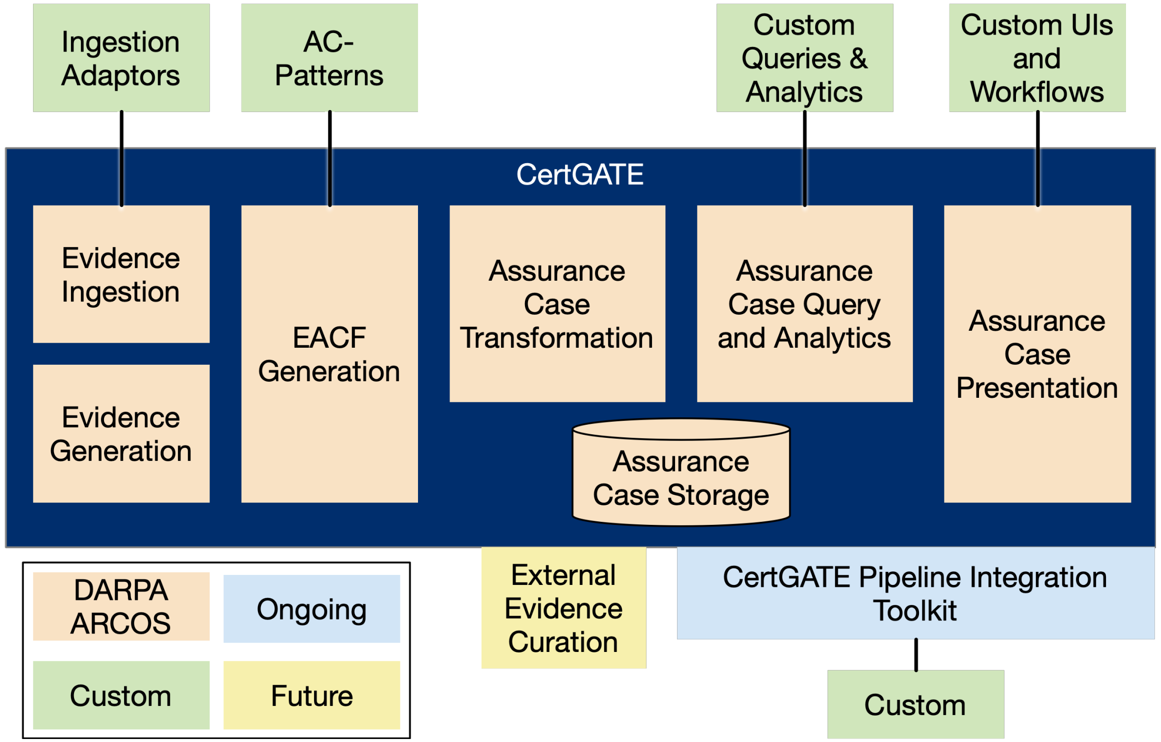
Digital Assurance Cases (DACs) are computational constructs designed to store, manage, and manipulate assurance information. **They are meant to be used by machines**.

“Assurance information”: not just data items/work products but also metadata (arguments, traceability linkages, linkages to configurations/baselines, etc.)

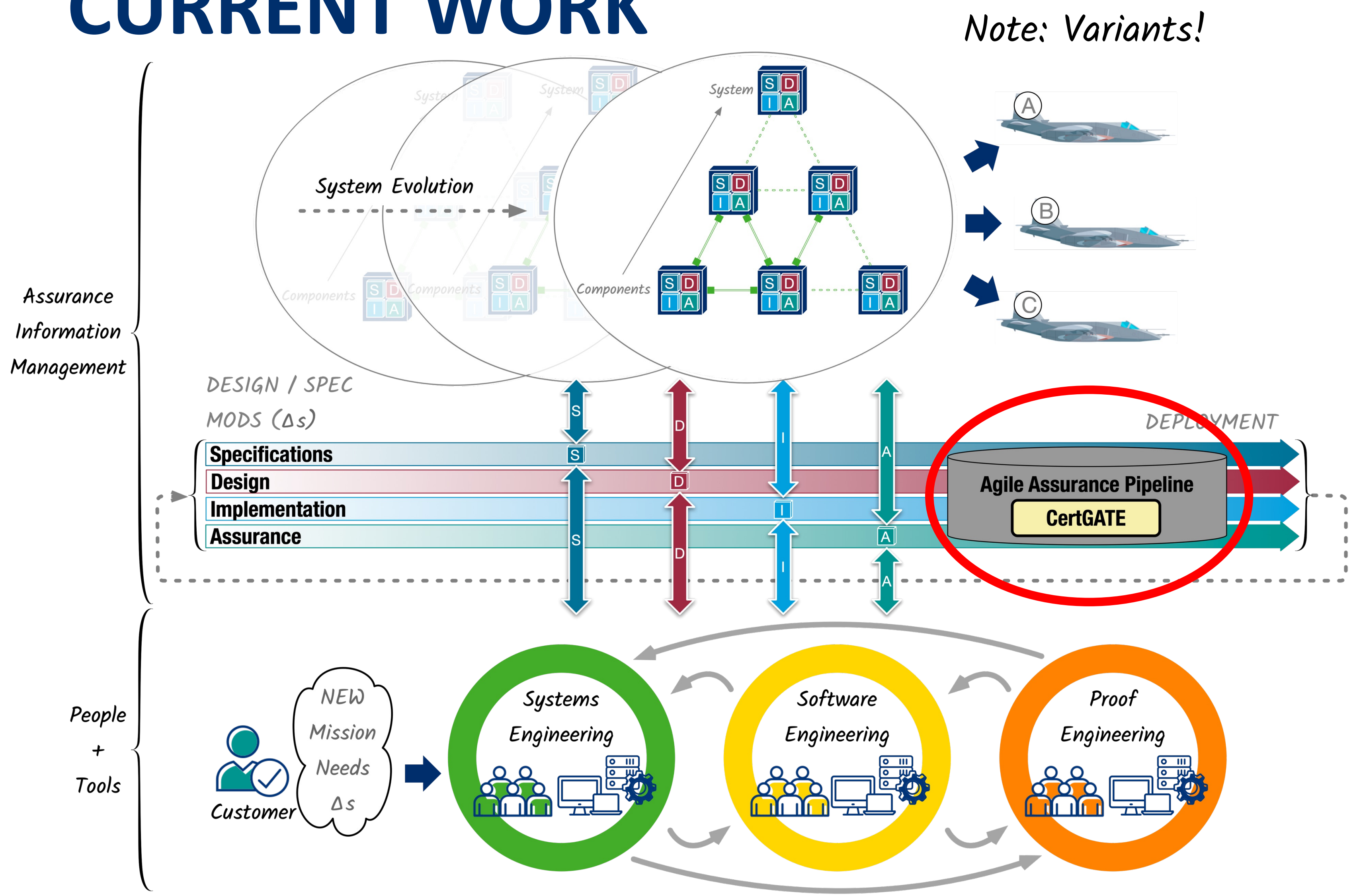
CertGATE

Separation of Concerns

- Separating concerns can help scale the application of assurance case technology for rapid certification
- It allows for development of best-in-class tools vs “jack of all trades” solutions
- Evidence Generation → build composable AC modules
- Assurance Case Construction
- Assurance Case Evaluation
- Assurance Case Presentation
- This can be achieved most effectively if all four concerns apply standard representations, e.g., SACM



CURRENT WORK



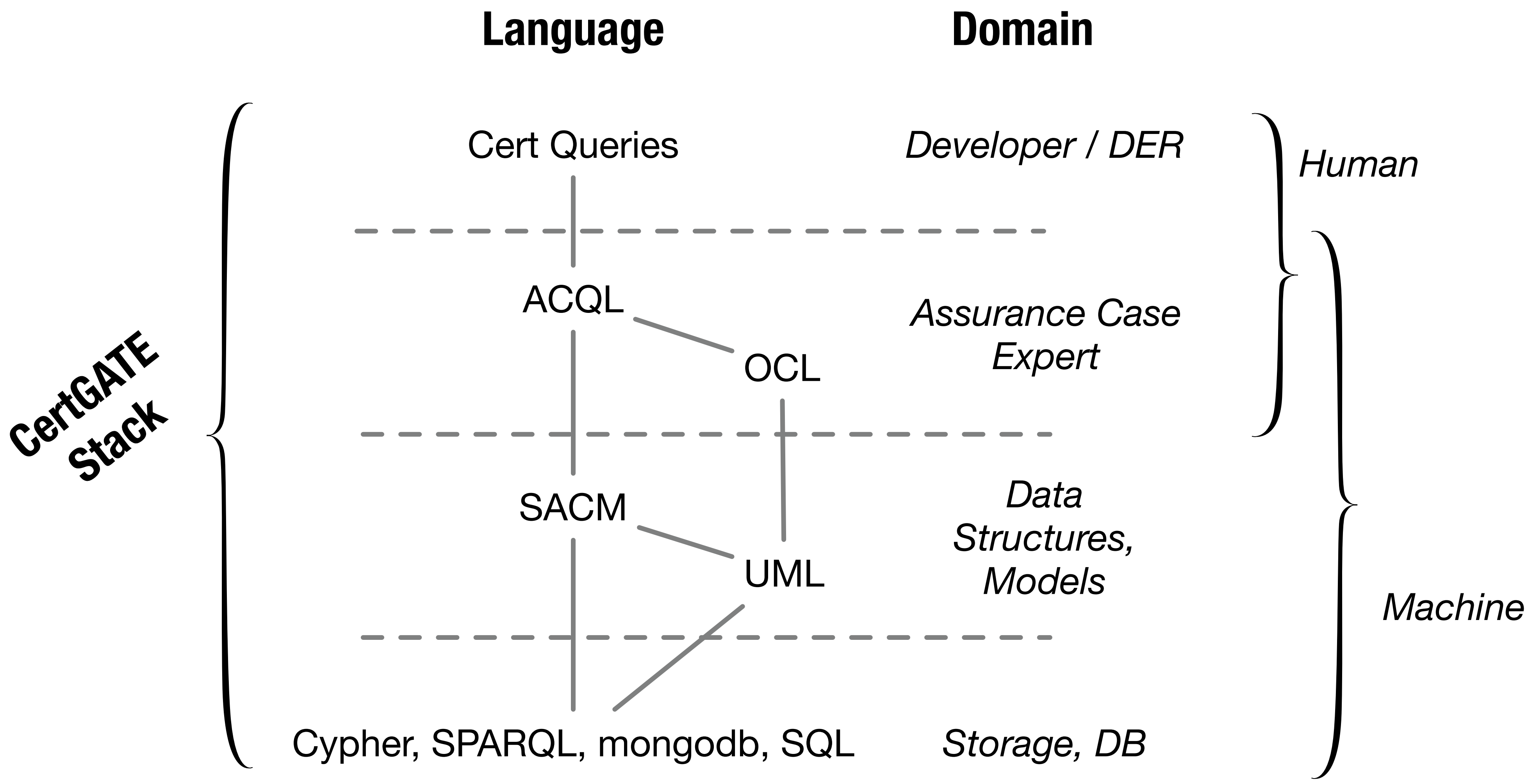
Four types of components

Specifications Design

Implementation Assurance

Type	Examples
S: Specifications	Requirements Contracts Properties Formal Specifications
D: Design	Models Architecture Diagrams
I: Implementation	Code Binaries Configuration files
A: Assurance	Proofs Tests Analysis results Inspection results

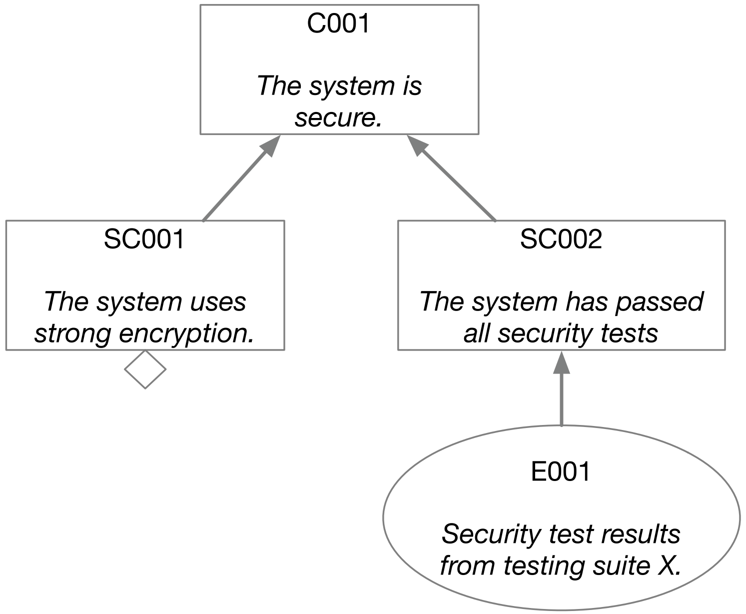
Assurance Case Query Language



Simple Example

SACM pseudo-code

```
{
  "Claim": {
    "id": "C001",
    "description": "The system is secure.",
    "supportedBy": [
      {
        "SubClaim": {
          "id": "SC001",
          "description": "The system uses strong encryption.",
          "status": "undeveloped",
          "evidence": []
        }
      },
      {
        "SubClaim": {
          "id": "SC002",
          "description": "The system has passed all security tests.",
          "status": "developed",
          "evidence": [
            {
              "id": "E001",
              "description": "Security test results from testing suite X."
            }
          ]
        }
      }
    ]
  }
}
```



Verify that all subclaims have evidence

```
context AssuranceCase::hasAllRequiredEvidence(): Boolean
derive: self.Claim.supportedBy→forAll(subclaim | subclaim.evidence→notEmpty())
```

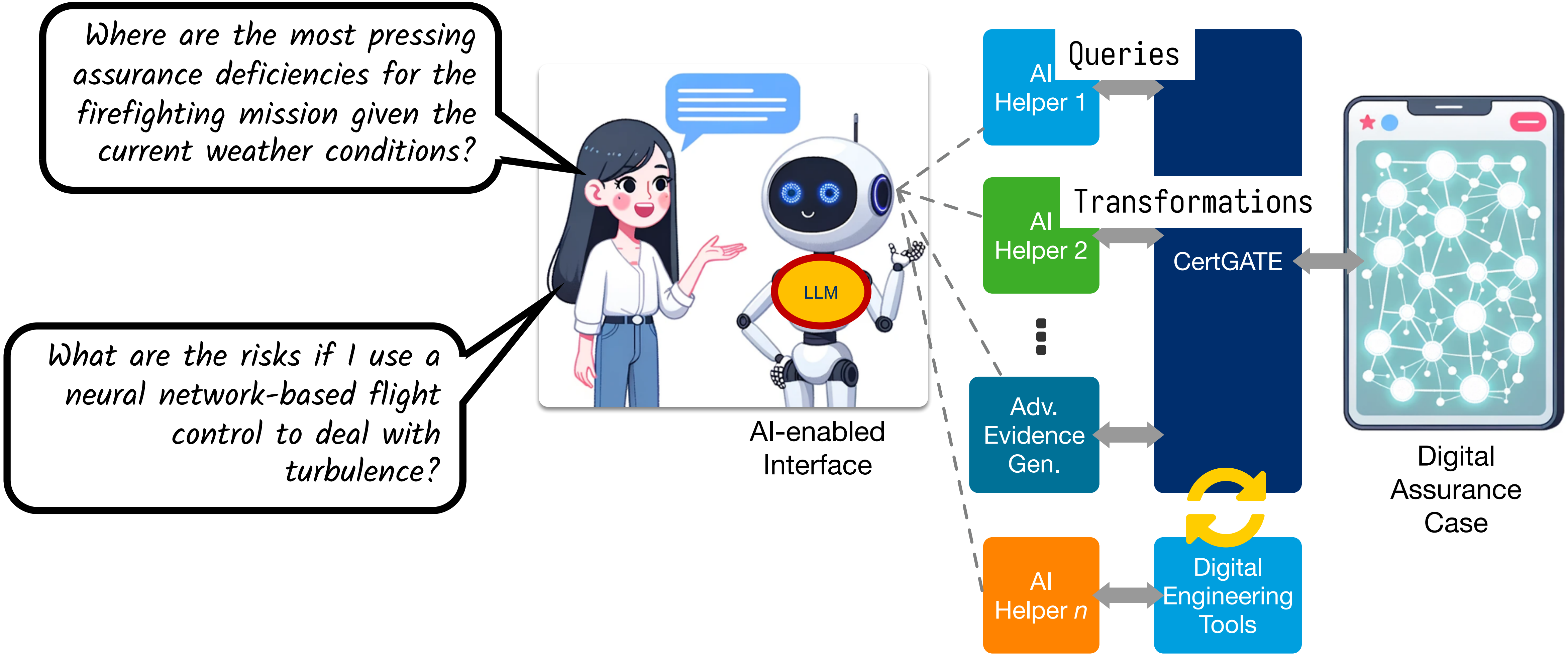
Result: False

Find IDs of undeveloped subclaims

```
context AssuranceCase::getUndevelopedSubclaimID(): Sequence(String)
derive: self.Claim.supportedBy→select(subclaim | subclaim.status = 'undeveloped').id
```

Result: SC001

Obvious Next Step



Summary So Far

Original intent of assurance cases

– **Capture our understanding** of the system’s ability to satisfy functional and non-functional properties and evidence supporting the confidence in these beliefs

– Make all assumptions, arguments, and justifications explicit **such that another human being can evaluate a “case”** supporting claims about these properties

Additional demands due to complexity and agility

– How do we capture information such that humans can arrive at conclusions with the help of machines more effectively?

– How do we make this process part of the design, development, and operation adding minimal friction?

  - Two aspects of integration: process (automation) and information (representation)
  - Knowledge is dynamic: it needs to be accumulated!

This is a knowledge representation problem!

Enter Knowledge Graphs

Brief History

- Early foundations (1950s–1970s): early foundations
- Semantic Web and Ontologies (1980s–1990s)
- Google's Knowledge Graph (2012): "things not strings"
- Expansion in Use Cases, Advancements in Technology (2010s–Present)
- Examples today
  - Google Search, Imdb, Wikipedia, Thomson Reuters, Airbnb, Facebook, Elsevier, Etc.

George Washington

198 languages

Contents hide

Article Talk

Read View source View history Tools

(Top)
Early life (1732–1752)
Colonial military career (1752–1758)
Marriage, civilian, and political life (1755–1775)
Commander in chief (1775–1783)
Early republic (1783–1789)
Presidency (1789–1797)
Post-presidency (1797–1799)
Burial, net worth, and aftermath
Personal life
Slavery
Historical reputation and legacy
See also
Notes
References
Bibliography
Further reading
External links

From Wikipedia, the free encyclopedia

"General Washington" redirects here. For other uses, see General Washington (disambiguation) and George Washington (disambiguation).

George Washington (February 22, 1732 – December 14, 1799) was an American Founding Father, military officer, and politician who served as the first president of the United States from 1789 to 1797. Appointed by the Second Continental Congress as commander of the Continental Army in 1775, Washington led Patriot forces to victory in the American Revolutionary War and then served as president of the Constitutional Convention in 1787, which drafted and ratified the Constitution of the United States and established the U.S. federal government. Washington has thus become commonly known as the "Father of his Country".

Washington's first public office, from 1749 to 1750, was as surveyor of Culpeper County in the Colony of Virginia. He subsequently received military training and was assigned command of the Virginia Regiment during the French and Indian War. He was later elected to the Virginia House of Burgesses and was named a delegate to the Continental Congress in Philadelphia, which appointed him commander-in-chief of the Continental Army. Washington led American forces to a decisive victory over the British in the Revolutionary War, leading the British to sign the Treaty of Paris, which acknowledged the sovereignty and independence of the United States. He resigned his commission in 1783 after the conclusion of the Revolutionary War.

Washington played an indispensable role in adopting and ratifying the Constitution, which replaced the Articles of Confederation in 1789. He was then twice elected president by the Electoral College unanimously. As the first U.S. president, Washington implemented a strong, well-financed national government while remaining impartial in a fierce rivalry that emerged between cabinet members Thomas Jefferson and Alexander Hamilton. During the French Revolution, he proclaimed a policy of neutrality while additionally sanctioning the Jay Treaty. He set enduring precedents for the office of president, including republicanism, a peaceful transfer of power, the use of the title "Mr. President", and the two-term tradition. His 1796 farewell address became a preeminent statement on republicanism in which he wrote about the importance of national unity and the dangers that regionalism, partisanship, and foreign influence pose to it.

George Washington

1st President of the United States

In office
April 30, 1789 – March 4, 1797

Vice President	John Adams
Preceded by	*Office established*
Succeeded by	John Adams
Additional offices	[show]
Personal details	
Born	February 22, 1732 [O.S. February 11, 1731] Popes Creek, Virginia Colony, British America
Died	December 14, 1799 (aged 67) Mount Vernon, Virginia, U.S.
Resting place	Mount Vernon, Virginia 38°42′28.4″N 77°05′09.9″W

Knowledge Card

KGs are used widely and supporting technologies continue evolving

KG Definition

“A graph of data intended to **accumulate** and **convey** knowledge of the world or a portion of the world. They are designed to support sophisticated reasoning and to be **easily interpreted by humans and computers**. Knowledge graphs are typically constructed by integrating **data from many different sources** and are often used in applications where **complex reasoning and inferences** are required” (Aidan Hogan et al. “Knowledge Graphs Survey”, 2021)

The overlap between KGs and Assurance Cases is remarkable!

Advantages of KGs for Representing ACs

- Already co-located with other information that has graph (of graphs) structure, e.g.: requirements traceability, testing, terminologies, configurations control, version control, etc.
- Connectivity with other graphs enriches the semantics of assurance cases (“things not strings,” avoidance of “word salads” and “letter soups”)
- Ideally suited for ingestion of raw information. Most industry-relevant use cases are “brown-field.” There is already a big collection of artifacts, documents, code, etc. that need to be reused.
- Powerful automated reasoning, querying, and inferencing.
- We can “piggy-back” on continuously evolving technologies widely used in other sectors of industry → high performance, reasonable cost, demonstrated scalability, security, privacy, etc.

LOCKHEED MARTIN

Future

KGs ❤️ LLMs

- Enhanced knowledge and factuality
  - Accurate and up-to-date information
- Improved contextual understanding
  - Entity recognition and disambiguation
  - Richer context
- Natural language interaction for ease of use
- Flexibility and personalization for reviews, inspection, and walkthroughs

Use of LLMs in conjunction with KGs will grow creating a richer substrate for AC reasoning!

Conclusions

We've started exploring the amenities of the ski resort. We'd like to meet other explorers.

LOCKHEED MARTIN

LOCKHEED MARTIN