Priority Quality Attributes for Engineering AI-enabled Systems

Lena Pons

Ipek Ozkaya

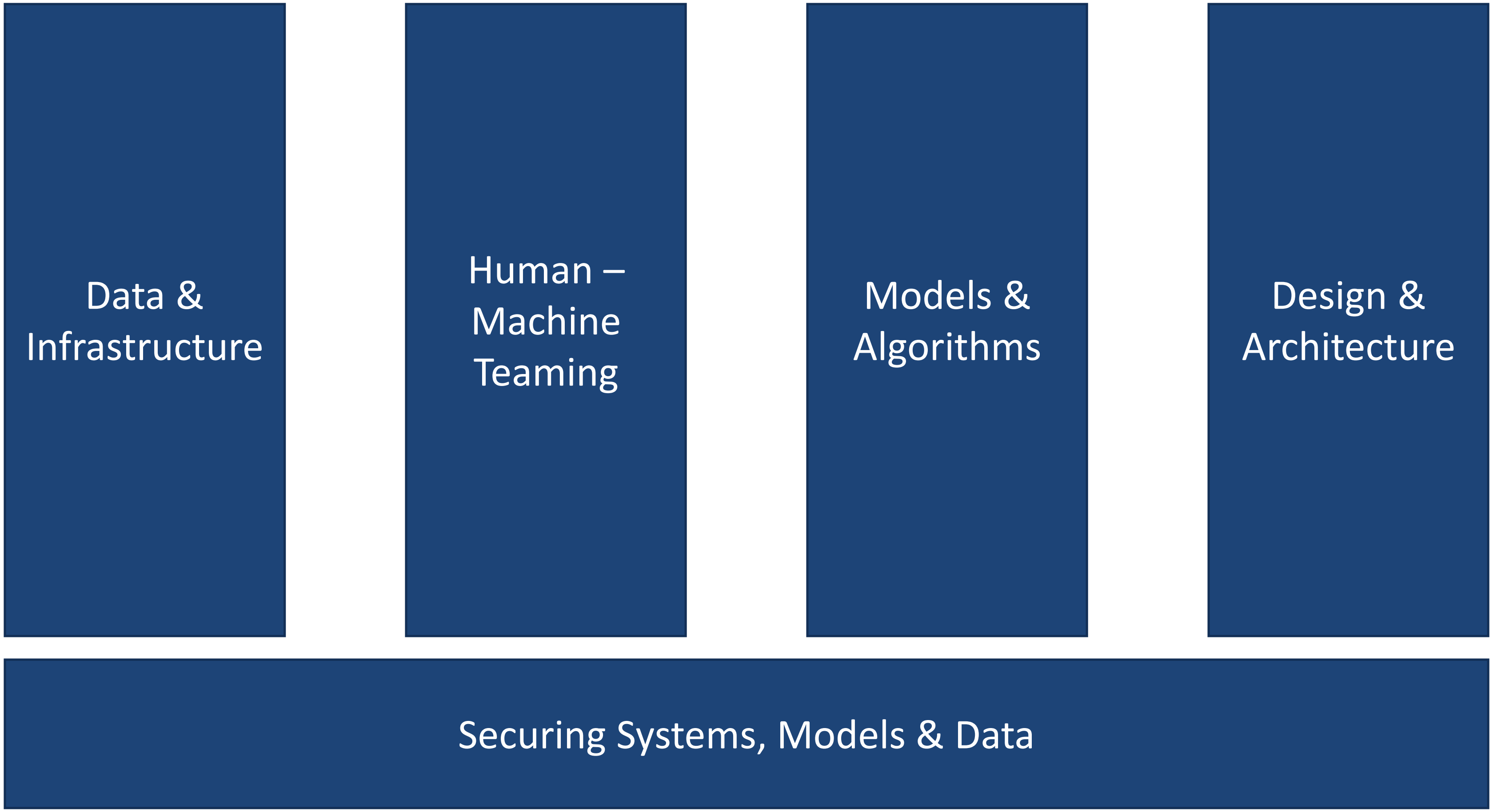
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Document Markings

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM19-1166

Cross-Cutting AI/ML Challenges



Data & Infrastructure

Human – Machine Teaming

Models & Algorithms

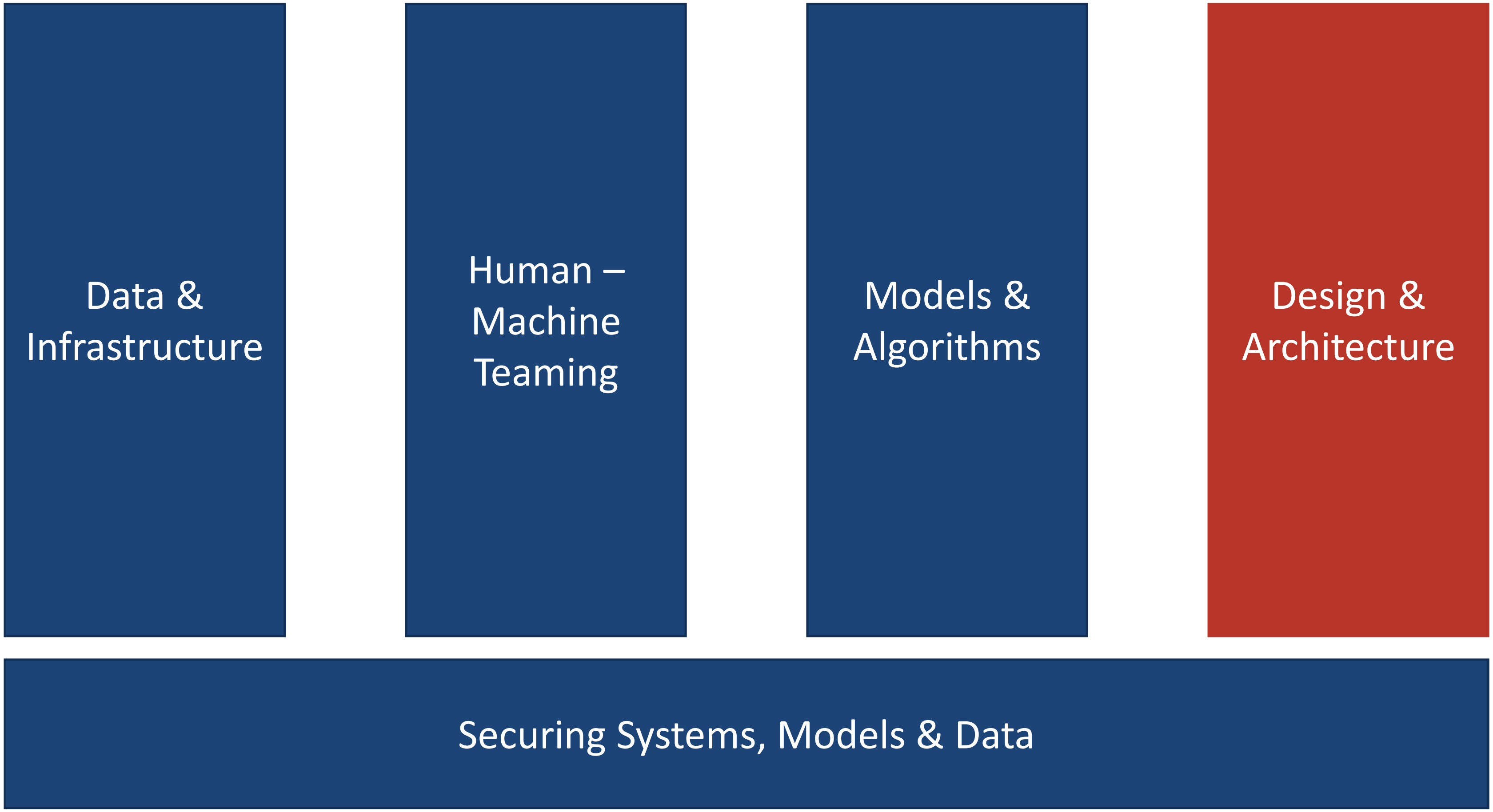
Design & Architecture

Securing Systems, Models & Data

Engineering AI-enabled Systems

- AI systems are built of software
- Engineering an AI-enabled system poses some challenges that are distinct from ‘conventional software’
- AI-enabled systems are not a monolith – e.g. neural network methods vs. regression based methods
- The interaction between software and data touches all of the challenges and architecture considerations we will discuss
- We need new methods and architecture solutions to design AI-enabled systems that can be confidently deployed in public sector context

Architecting AI-enabled Systems



Data & Infrastructure

Human – Machine Teaming

Models & Algorithms

Design & Architecture

Securing Systems, Models & Data

Quality Attributes

Quality attributes are properties of work products or goods by which stakeholders judge their quality.

Software cannot be designed to optimize for all quality attributes

- Software engineers must select which attributes are most important for the stakeholder needs
- Stakeholder needs will vary with the context in which a software system is deployed

The degree to which a software system meets its quality attribute requirements depends on its architecture.

- Architectural decisions are made to promote various quality attributes.

How can we transfer architecture thinking to AI-enabled system design

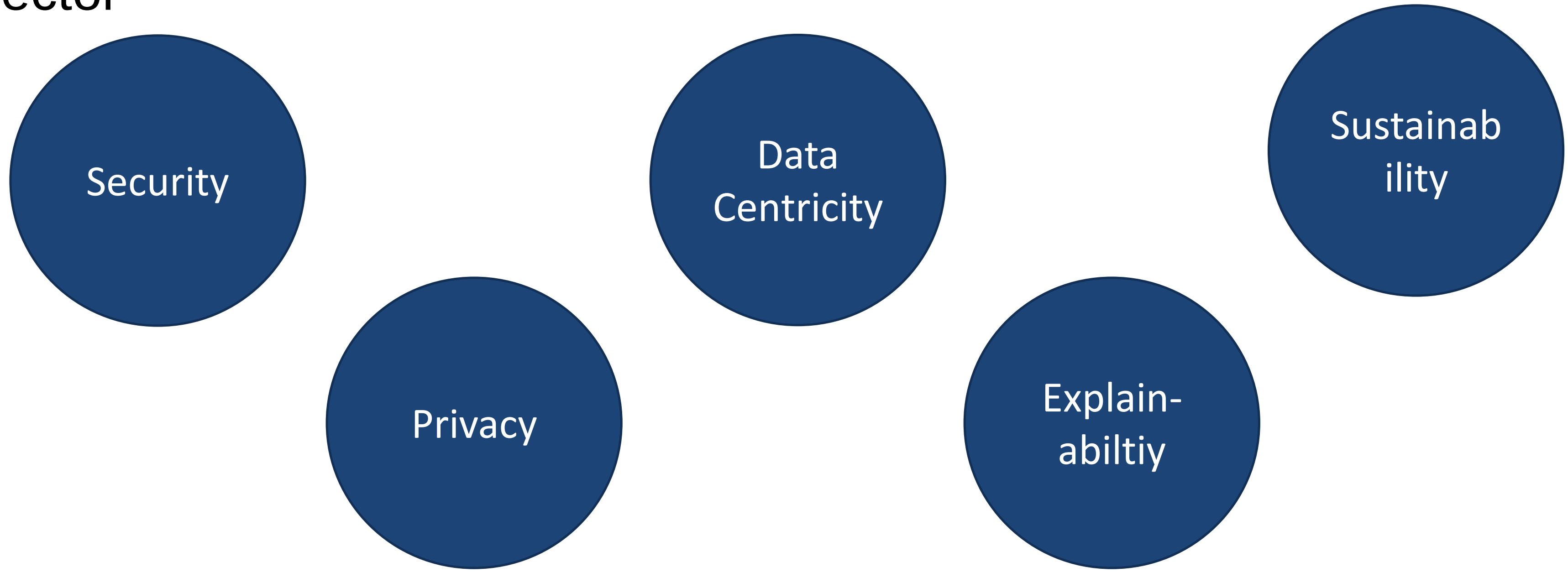
Architecture permits/precludes the *achievement of a system's desired quality attributes*. The strategies for achieving these requirements require thinking about the structure and behavior of the system.

If you desire…	you need to pay attention to at a minimum…
high performance	minimizing the frequency and volume of inter-element communication
modifiability	limiting interactions between elements
security	managing and protecting inter-element communication
availability	the properties and behaviors that elements must have and the mechanisms you will employ to address fault detection, fault prevention, and fault recovery
extensibility	limiting interactions between elements and isolate data types, abstract common services

Public Sector Context

Systems deployed in the public sector have a higher sensitivity to some of these quality attributes than systems deployed elsewhere

We identify 5 priority quality attributes for AI systems in the public sector



Security

Privacy

Data Centricity

Explainabiltiy

Sustainability

Engineering AI Systems

An AI-enabled software system is a software system with one ore more AI component(s) which need to be developed, deployed, and sustained along with the other software and hardware elements of the system.

<put some kind of diagram here

Upstream software -> glue -> model -> glue -> downstream software>

The integration of an AI component makes some parts of system assurance harder

<data, verification, security & privacy, intepretability/ explainability, & #5>

Security

AI engineering challenge:

AI-enabled systems introduce new attack surfaces.

Quality attribute reasoning recommendations:

- Design systems to
- Decouple model changes from the rest of the system changes
- Build-in to your architecture modifiability to deal with deploying retrained models

Privacy

AI engineering challenge:

Data and modeling are tightly coupled in AI systems, which makes privacy protections difficult or impossible to implement.

Quality attribute reasoning recommendations:

- Understand privacy implications of a data collection and a data collection’s interaction with other accessible data collections
- Provide users with information about their risk from information collection
- Acknowledge interdependency of privacy and security

Data Centricity

AI engineering challenge:

Data influences every aspect of AI system design and the impact of data changes over time may not be adequately addressed by the software update cycle

Quality attribute reasoning recommendations:

- Architect systems with uncertainty, availability and scalability of data in mind
- Design systems to allow for monitoring of performance as data changes over time
- Plan for data changes to trigger out of cycle updates to models

Explainability

AI engineering challenge:

The output of AI models is not always directly interpretable to a human user. Frequently interpreting outputs relies on an individual being highly trained in statistics and another discipline

Quality attribute reasoning recommendations:

- Decouple model changes from the rest of the system changes
- Build-in to your architecture modifiability to deal with deploying retrained models
- Provide multiple modes of communicating output to help combat communication challenges

Sustainability

AI engineering challenge:

Rates of change that impact software and AI/ML components differently, creating inconsistencies when software needs to be deployed, evolved, or replaced

Quality attribute reasoning recommendations:

- Express rate of change requirements as architectural concerns
- Decouple model changes from the rest of the system changes
- Build-in to your architecture modifiability to deal with deploying retrained models

Looking Ahead

AI-enabled systems pose new challenges in explainability, accuracy, security and ethics,

- Architectural thinking helps improve our understanding of these concerns.

As research progresses in developing analysis and conformance approaches, locking approved algorithms, defining conservative update cycles will support early wins in deploying AI systems.

Repeatable practice will emerge as we create consistent vocabulary around expressing and analyzing for related design concerns in AI systems.