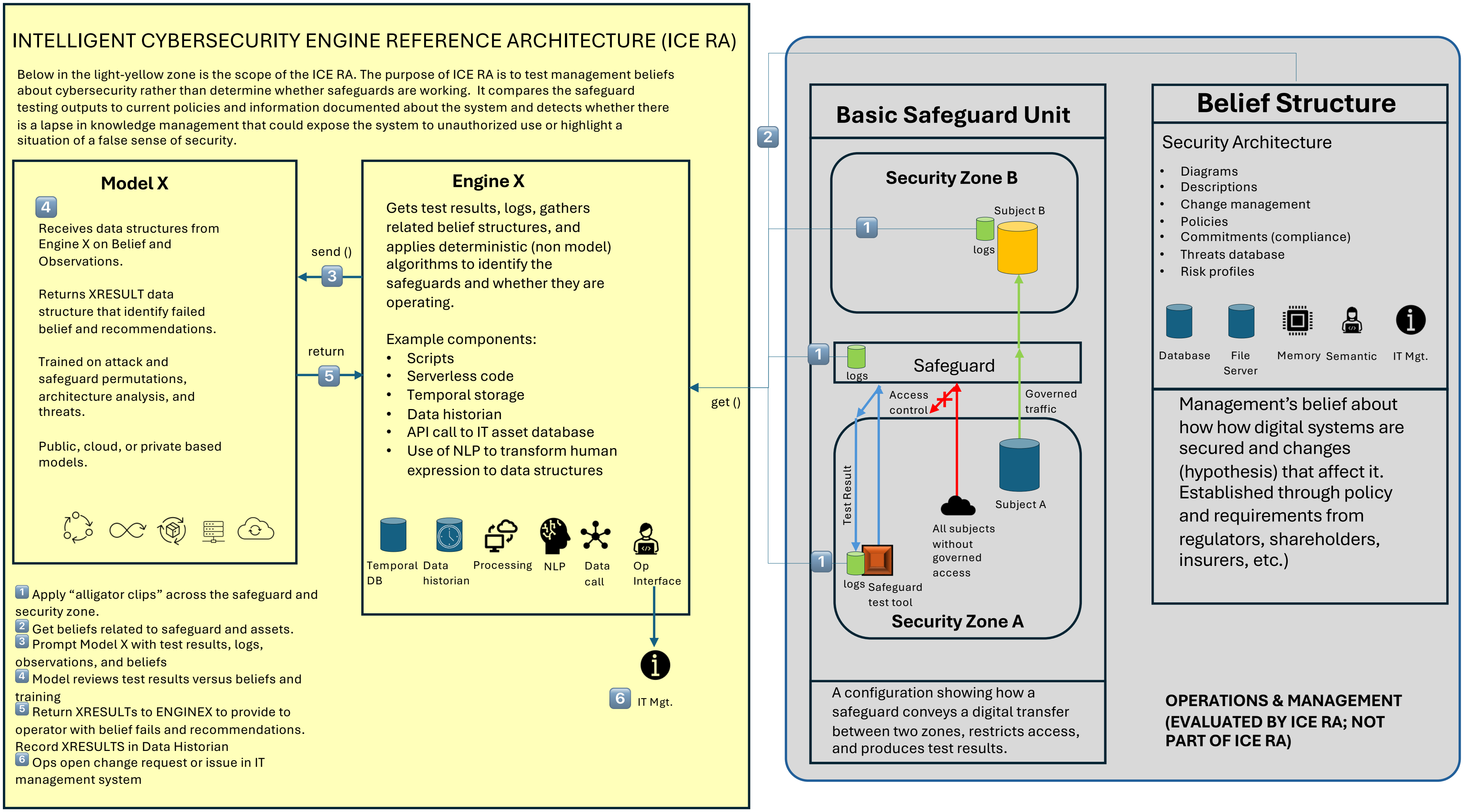
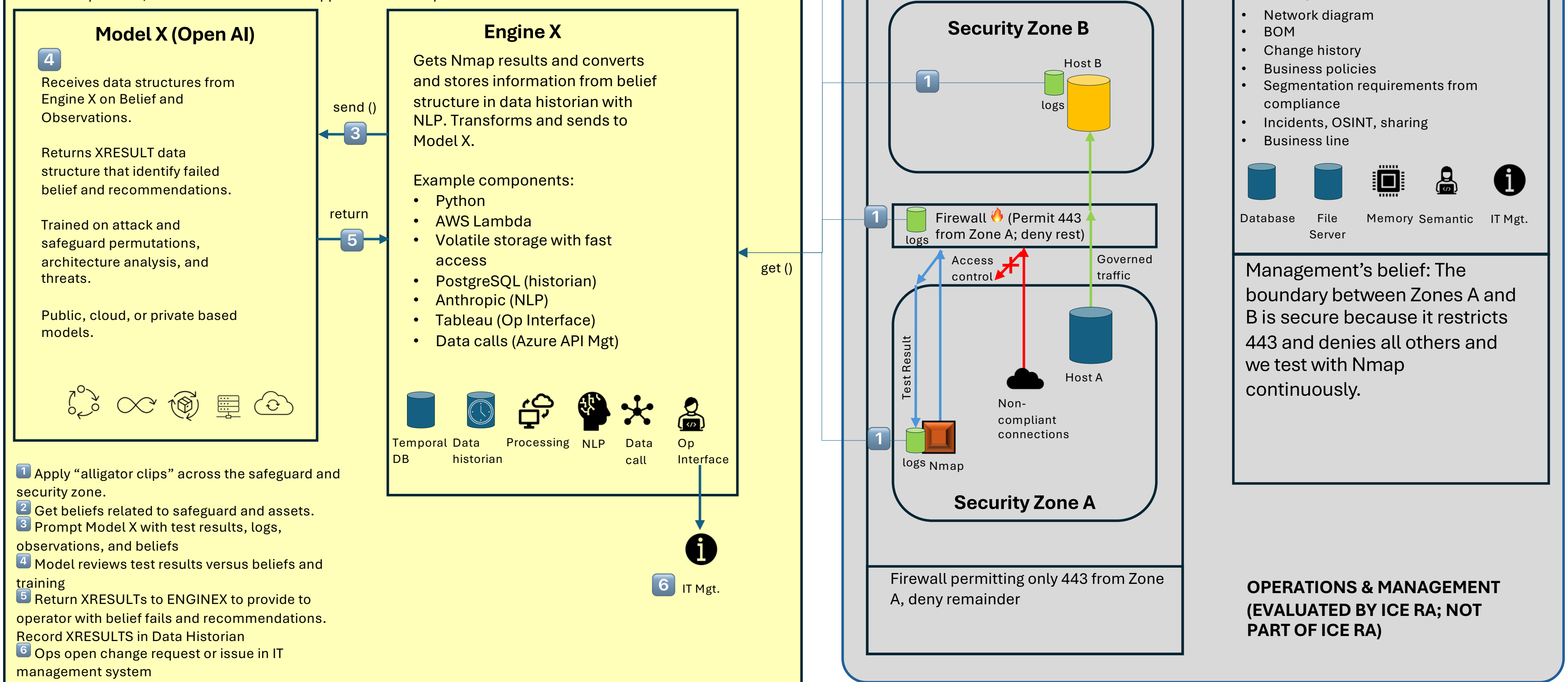
Intelligent Cybersecurity Engine Reference Architecture - Contents

- ICE RA Overview: Conceptual layout of the main ICE RA components, steps and data flow, and the simplest test environment.
- Firewall case: Simple steps and data flow for two security zones and a firewall.
- “Alligator clip” view + XRESULTS – simple view that shows how XRESULTS are yielded from ICE RA. Includes “questions” answered by XRESULTS.


INTELLIGENT CYBERSECURITY ENGINE REFERENCE ARCHITECTURE (ICE RA)
Below in the light-yellow zone is the scope of the ICE RA. The purpose of ICE RA is to test management beliefs about cybersecurity rather than determine whether safeguards are working. It compares the safeguard testing outputs to current policies and information documented about the system and detects whether there is a lapse in knowledge management that could expose the system to unauthorized use or highlight a situation of a false sense of security.
Model X
4
Receives data structures from Engine X on Belief and Observations.
Returns XRESULT data structure that identify failed belief and recommendations.
Trained on attack and safeguard permutations, architecture analysis, and threats.
Public, cloud, or private based models.
send ()
3
return
5
Engine X
Gets test results, logs, gathers related belief structures, and applies deterministic (non model) algorithms to identify the safeguards and whether they are operating.
Example components:
• Scripts
• Serverless code
• Temporal storage
• Data historian
• API call to IT asset database
• Use of NLP to transform human expression to data structures
Temporal DB
Data historian
Processing
NLP
Data call
Op Interface
6
IT Mgt.
get ()
1 Apply “alligator clips” across the safeguard and security zone.
2 Get beliefs related to safeguard and assets.
3 Prompt Model X with test results, logs, observations, and beliefs
4 Model reviews test results versus beliefs and training
5 Return XRESULTs to ENGINEX to provide to operator with belief fails and recommendations. Record XRESULTS in Data Historian
6 Ops open change request or issue in IT management system
2
Basic Safeguard Unit
Security Zone B
1
Subject B
logs
1
logs
Safeguard
Access control
Governed traffic
Test Result
All subjects without governed access
Subject A
1
logs
Safeguard test tool
Security Zone A
A configuration showing how a safeguard conveys a digital transfer between two zones, restricts access, and produces test results.
Belief Structure
Security Architecture
• Diagrams
• Descriptions
• Change management
• Policies
• Commitments (compliance)
• Threats database
• Risk profiles
Database
File Server
Memory
Semantic
IT Mgt.
Management’s belief about how how digital systems are secured and changes (hypothesis) that affect it. Established through policy and requirements from regulators, shareholders, insurers, etc.)
OPERATIONS & MANAGEMENT (EVALUATED BY ICE RA; NOT PART OF ICE RA)

MVP CANDIDATE ZERO: ICE RA BASIC FIREWALL CASE

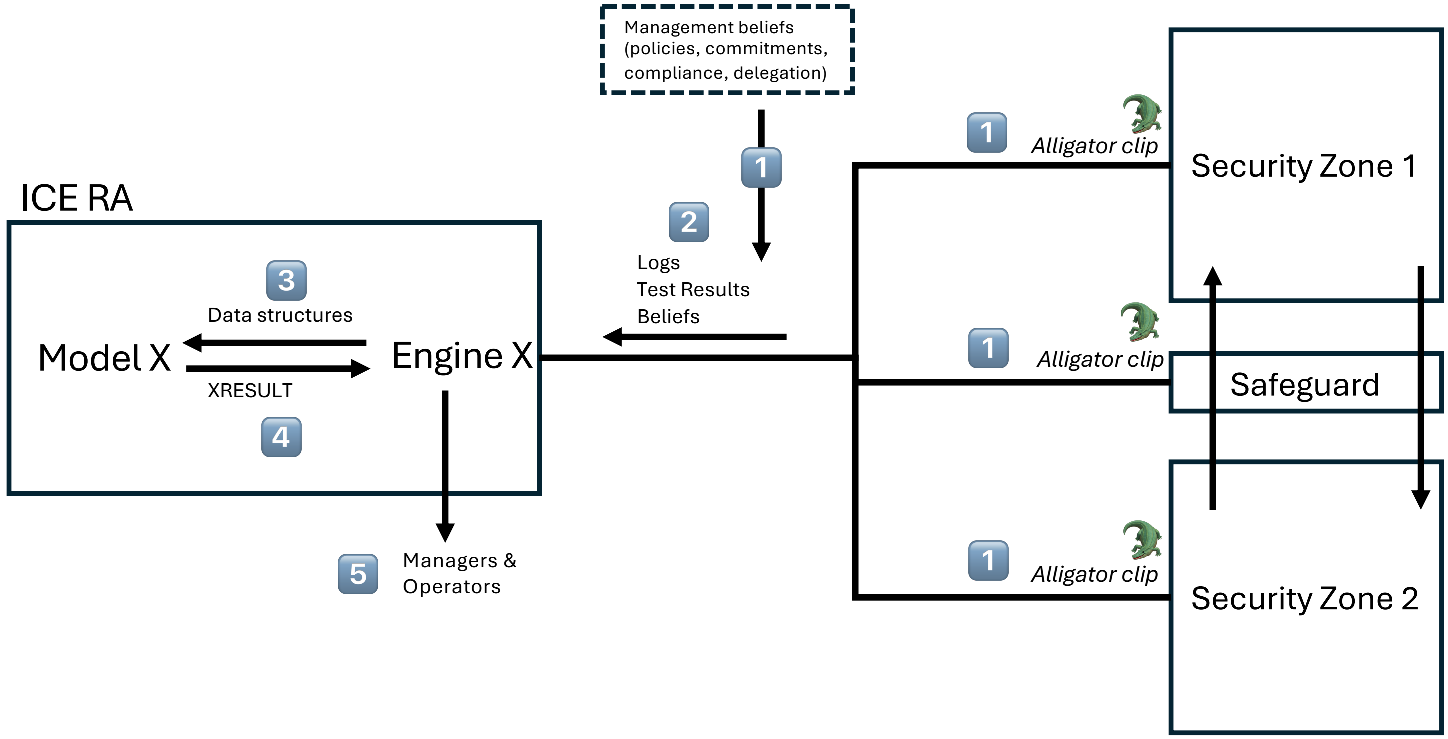
Below is a proposed basic implementation for ICE RA with proposed components. This is for the simplest case of a firewall separating two security zones. The purpose of this configuration is to determine whether the use of the firewall and Nmap testing supports management's beliefs about security, based on policy and commitments. In this case, the model might return something like "The testing does not support management's beliefs because the firewall configuration cannot confirm layer 7 authentication or unauthorized channels operating over 443". In other words, the existence of the firewall does not fully support the claim that the interface is secure even with Nmap testing. While this is a simple case, the value comes in more apparent with complex architectures.



XRESULTS Version 0

Below is the initial set of XRESULTS and implications returned by MODEL X because of reviewing management beliefs and test evidence.

1) Are environmental changes outpacing controls and testing?
2) Do the log results from the "alligator clips" support each other?
3) Is the evidence sufficient to support management belief?
4) Is the expected result attributable to intended configuration, or did the expected result appear despite a failed condition (false sense of security)
5) Are prohibited capabilities detectable?
6) Is the observed failure classified for purposes of analysis and history?
7) Is the observed behavior beyond the intent of management's policies?
8) Can we trace back drift between belief and behavior to the incident that caused or perpetuated it?
9) Are systems engineering practices susceptible to causing failures?
10) Is the cadence for monitoring acceptable given resource consumption and risk profile?
11) Is the belief correct enough to not "over-claim" security?
12) Are there indicators of indirect unauthorized signals crossing the security zone?



Note: XRESULT exists to identify drift between a properly working security apparatus and an environment that has changed due to disruptive circumstances: talent turnover, new technology, engineering changes, etc. and to identify root cause, effect on resources, testing cadence, & tool-asset mismatch (e.g. on-prem tools being used for cloud)