THE VALUE OF PERFORMANCE.
NORTHROP GRUMMAN

Northrop Grumman Cybersecurity Research Consortium (NGCRC)

Intelligent Autonomous Systems based on Data Analytics and Machine Learning

19 April 2018

Bharat Bhargava
Purdue University

Technical Champions: Jason Kobes, Jeffrey Ciocco, Will Chambers, Miguel Ochoa, Steve Seaberg, Peter Meloy, Paul Conoval, Jessica Trombley-Owens, Robert Pike, Brock Bose, Sam Shekar, Roderick Son

Intelligent Autonomous Systems

NORTHROP GRUMMAN

- According to **Wes Bush**, CEO of NGC, Autonomous Systems[1] should be
  - Able to perform complex tasks without or with limited ongoing connection to humans.
  - Cognitive enough to act without a human's judgment lapses or execution inadequacies.

- Intelligent Autonomous Systems (IAS) are characterized as highly **Cognitive**, effective in **Knowledge Discovery**, **Reflexive**, and **Trusted**,

[1]Wes Bush, Sept. 6, 2016. "The Exciting Future of Autonomous Systems" at KSU

Implemented Components of IAS

NORTHROP GRUMMAN

- **Cognitive Autonomy** & **Knowledge Discovery:**
  - Monitor and record system's activities (Data provenance and sequence of system calls)
  - Perform advanced analytics on provenance data, discover new patterns, and make predictions.
  - Deep learning based anomaly detection by analyzing sequence of system calls.
- **Reflexivity:**
  - Adapt to meet the mission objectives without disrupting the ongoing critical processes by incremental learning.
- **Trust:**
  - Provide consensus, verifiability, and integrity by using blockchain for storing provenance data.

Observations & Data from Experiments

NORTHROP GRUMMAN

- **Demo 1 (Cognitive Autonomy / Knowledge Discovery):**
  - Analytics over trusted provenance data to understand the current status of the system and take actions based on the result.
  - Performing aggregate analytics with data perturbation to protect the privacy of individual entities in IAS network.

Observations & Data from Experiments

NORTHROP GRUMMAN

- **Demo 2 (Reflexivity):**
  - Under anomalous operating contexts or attacks, the replica replacement design based on Combinatorial balanced incomplete modules, can take over the processing from primary module.
  - Replicas are updated with system states periodically (Update interval is determined through Bayesian inference of system's operating context).
  - Unused replicas are used for other processes, which makes the system to be faster and fault-tolerant.

Observations & Data from Experiments

NORTHROP GRUMMAN

- **Demo 3 (Trust):**
  - Scheme which guarantees integrity of provenance data is implemented
  - Capability to verify every transaction in IAS

THE VALUE OF PERFORMANCE.
NORTHROP GRUMMAN

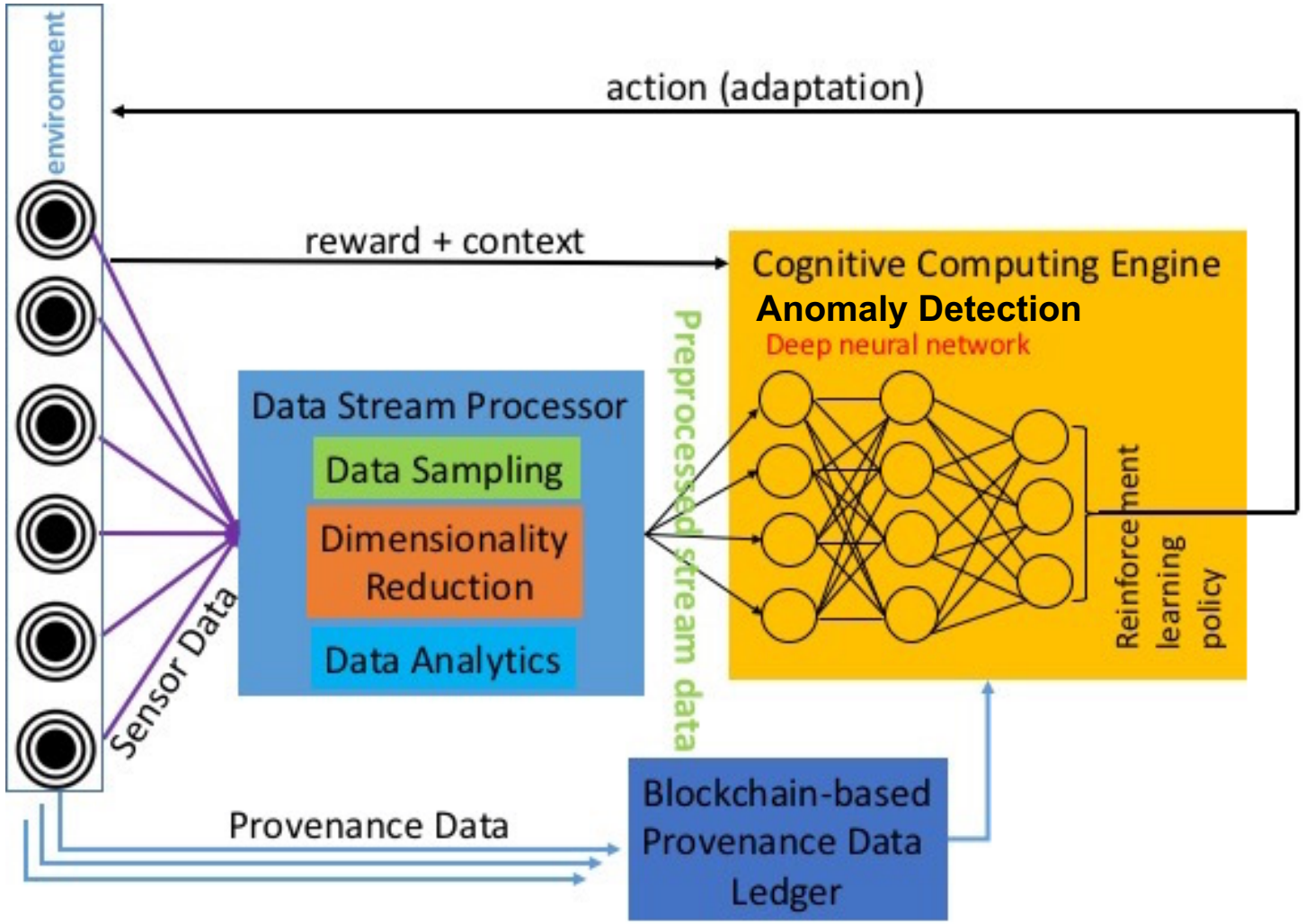
Reflexivity

Graceful Degradation Based on Machine Learning

CYBERSECURITY RESEARCH Consortium

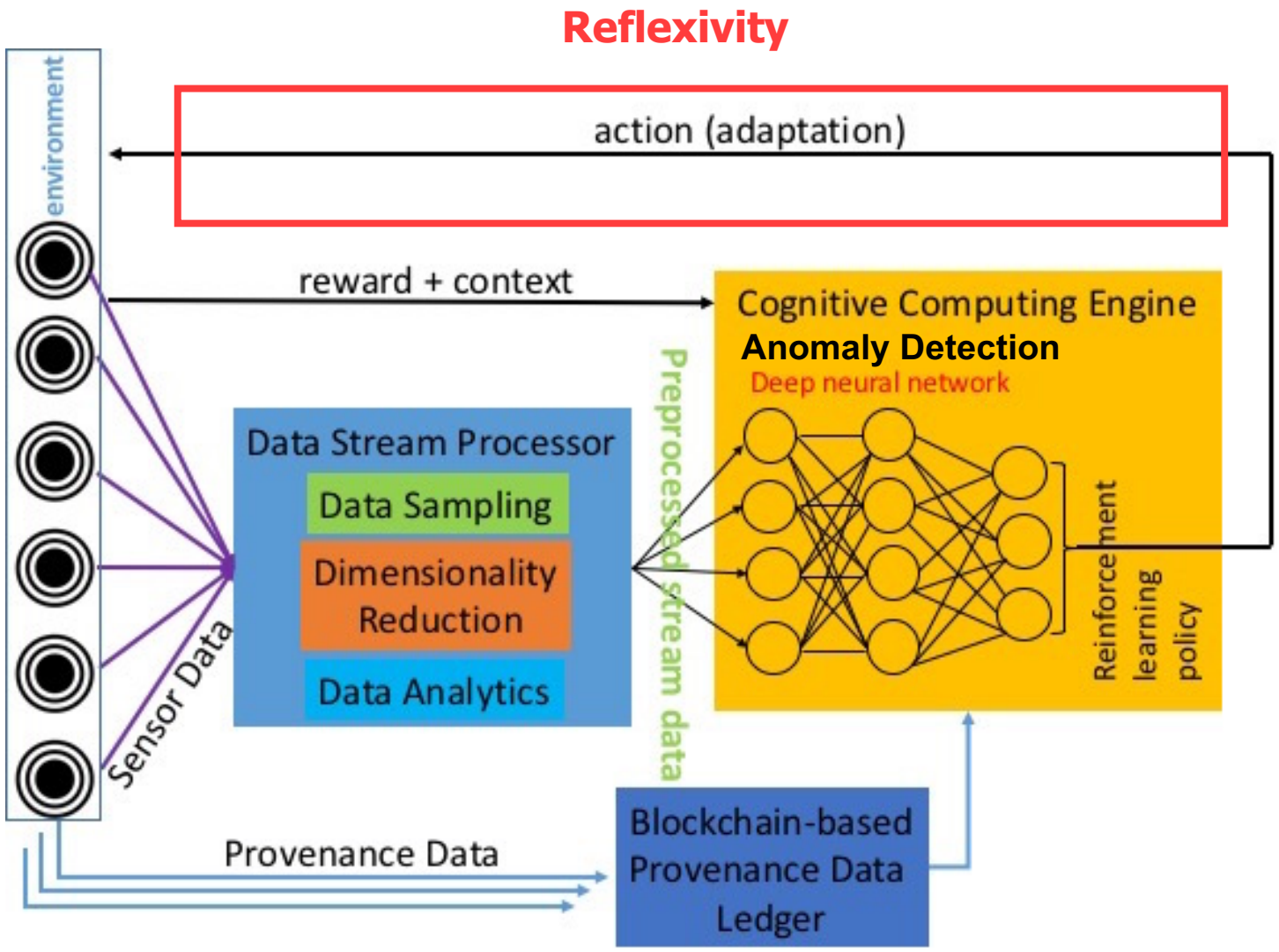
Comprehensive Architecture of IAS

NORTHROP GRUMMAN

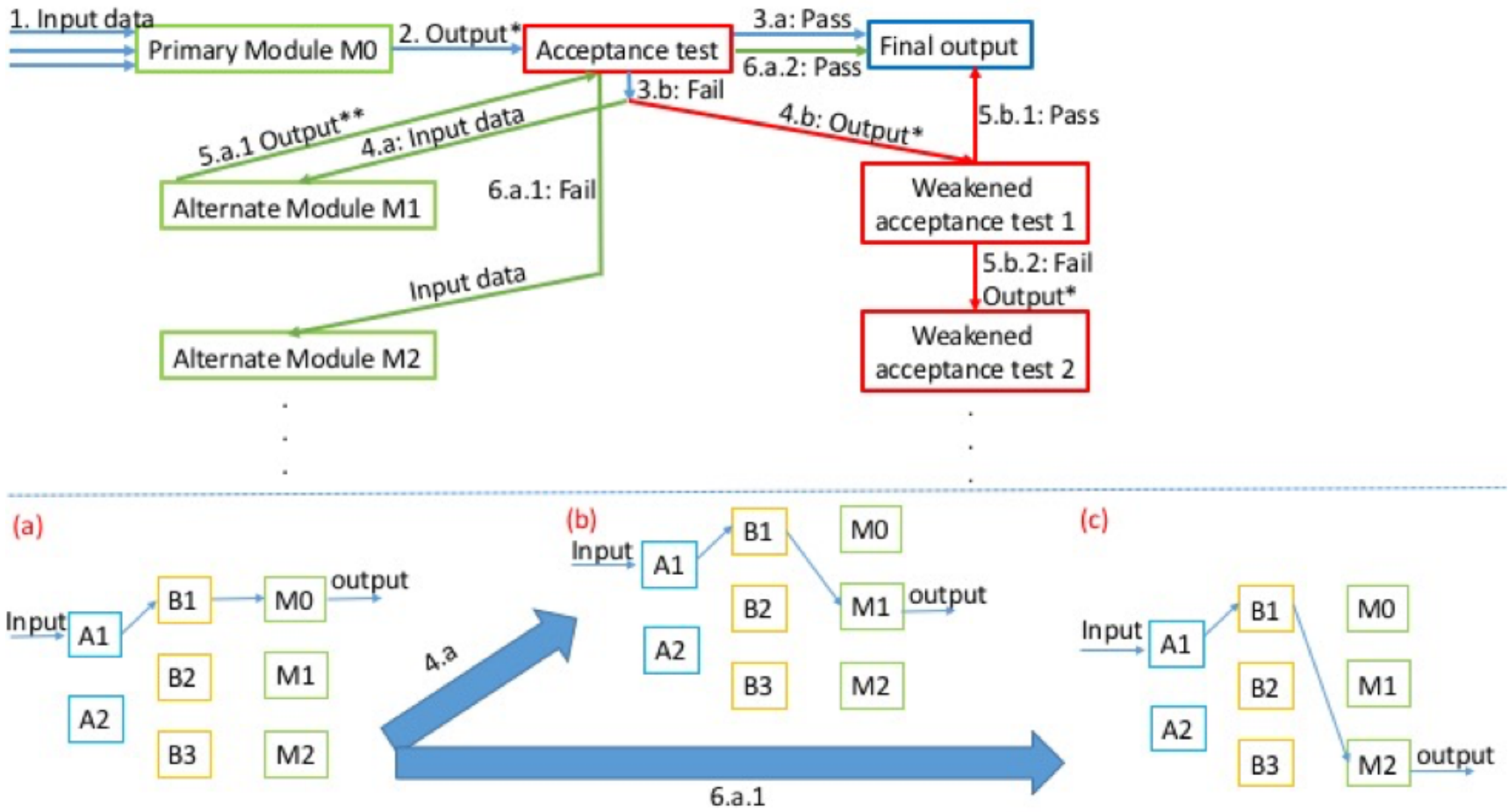


Comprehensive Architecture of IAS

NORTHROP GRUMMAN



Generic Model of Dynamic Adaptation



Problem Statement

NORTHROP GRUMMAN

Given a smart cyber system operating in a distributed computing environment, it should be able to:

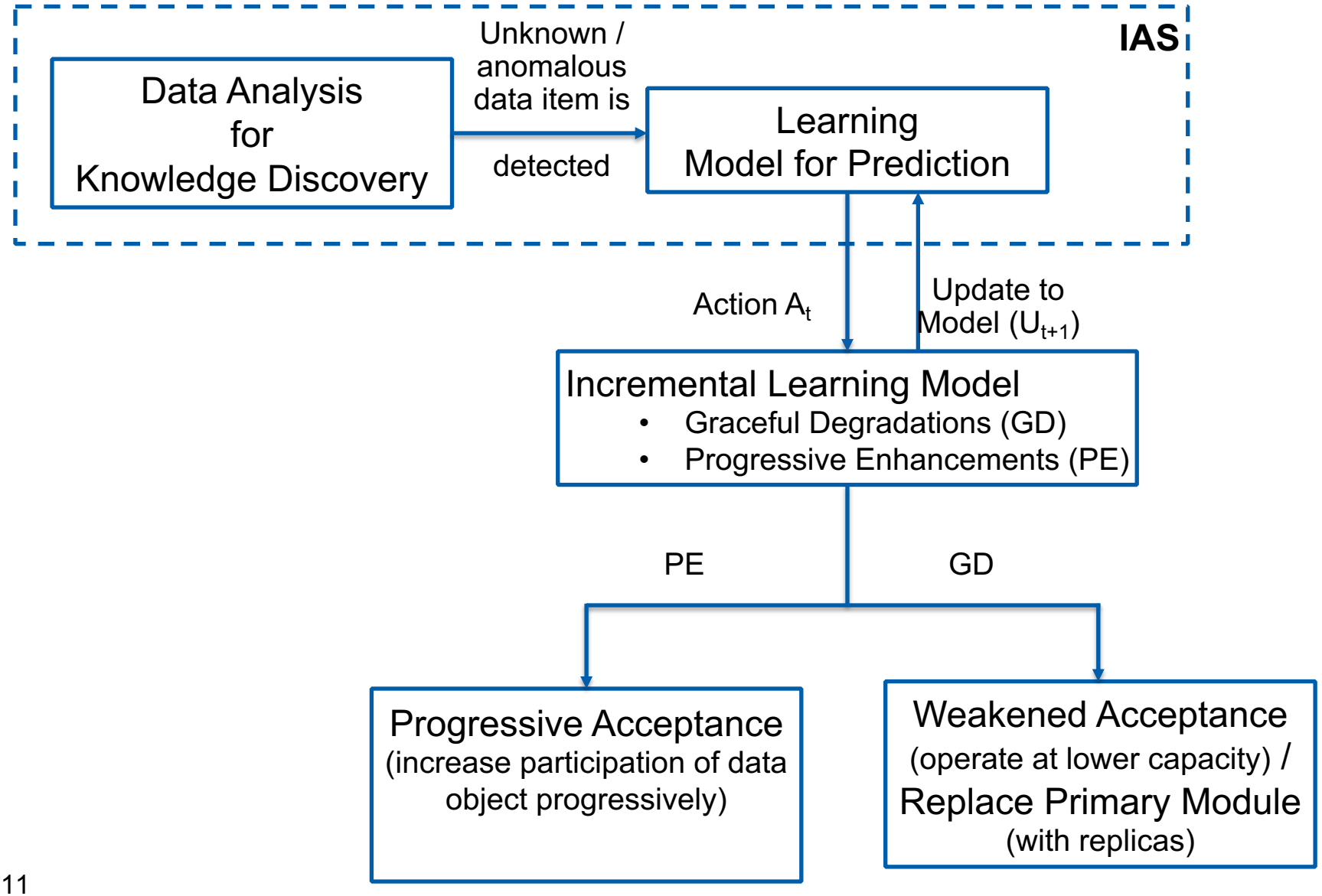
1. Replace anomalous/underperforming modules

2. Swiftly adapt to changes in context

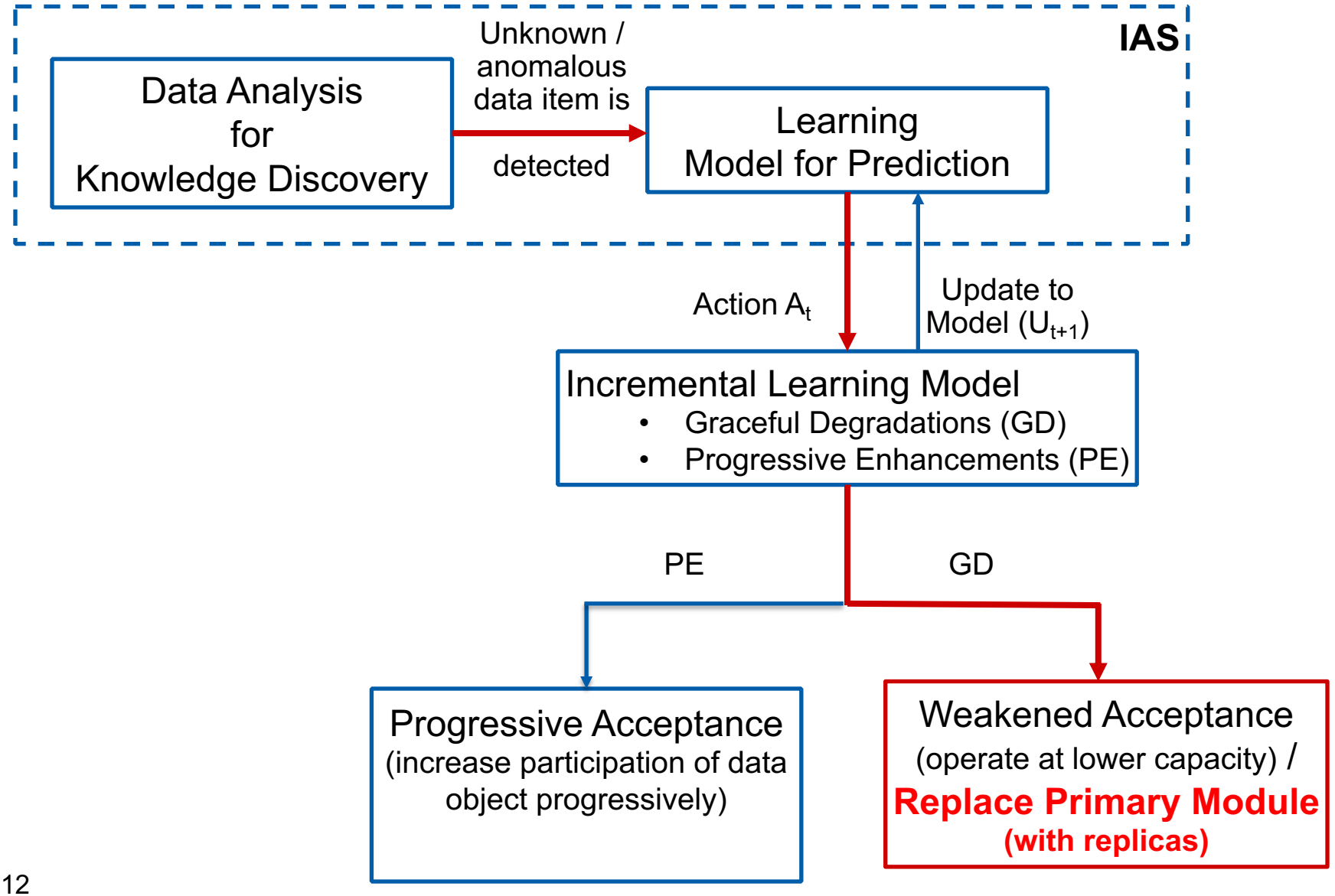
3. Achieve continuous availability even under attacks and failures[4].

[4]**Thomas E. Vice, Corporate VP of NGC**. Sep. 06, 2016. “"Future of Advanced Trusted Cognitive Autonomous Systems,” at Purdue University

Reflexivity Workflow



Reflexivity Workflow: Graceful Degradations



Graceful Degradations: Replica Replacement Technique

NORTHROP GRUMMAN

Replica replacement by Combinatorial Balanced-block Designs:

- Combinatorial Structure is a subset satisfying certain conditions.
- Each block contains systems and their replicas that are mathematically distributed.
- The systems and their replicas in the distributed blocks are strategically connected to receive updates from primary modules.
- Resources are mathematically balanced, enabling scalable designs for the systems.

Combinatorial Balanced-block Structure

NORTHROP GRUMMAN

- It is distributed environment with
  - A set ***Z*** consisting ***N*** systems
  - ***M*** distributed blocks consisting of
  - Subset of N system of size of ***R***
  - Each system in set N appears exactly in ***C*** subsets
  - Each pair in N systems appears exactly in Δ subsets.

Combinatorial Balanced-block Structure: Our implementation

NORTHROP GRUMMAN

- It is distributed environment with
  - A set ***Z*** consisting ***7*** systems
  - ***M = 7*** distributed blocks consisting of
  - Subset of Z of size of ***R = 3***
  - Each system in Z appears exactly in ***C = 3*** subsets (3 replicas)
  - Each pair in Z appears exactly in Δ **= 1** subsets.

(7, 7, 3, 3, 1)-configuration

NORTHROP GRUMMAN

- 7 systems $\{S_1, S_2, S_3, S_4, S_5, S_6, S_7\}$
- 7 Distributed Autonomous Blocks (DABs) each with 3-system subset

$$DAB_1 = \{S_1, S_5, S_7\},\ DAB_2 = \{S_1, S_2, S_6\},$$
$$DAB_3 = \{S_2, S_3, S_7\},\ DAB_4 = \{S_1, S_3, S_4\},$$
$$DAB_5 = \{S_2, S_4, S_5\},\ DAB_6 = \{S_3, S_5, S_6\},$$
$$DAB_7 = \{S_4, S_6, S_7\}.$$

(7, 7, 3, 3, 1)-configuration

NORTHROP GRUMMAN

- 7 systems $\{S_1, S_2, S_3, S_4, S_5, S_6, S_7\}$
- 7 Distributed Autonomous Blocks (DABs) each with 3-system subset
- Each system appears in 3 DABs (Say, S_6)

$$DAB_1 = \{S_1, S_5, S_7\},\ \mathbf{DAB_2} = \{S_1, S_2, \mathbf{S_6}\},$$
$$DAB_3 = \{S_2, S_3, S_7\},\ DAB_4 = \{S_1, S_3, S_4\},$$
$$DAB_5 = \{S_2, S_4, S_5\},\ \mathbf{DAB_6} = \{S_3, S_5, \mathbf{S_6}\},$$
$$\mathbf{DAB_7} = \{S_4, \mathbf{S_6}, S_7\}.$$

(7, 7, 3, 3, 1)-configuration

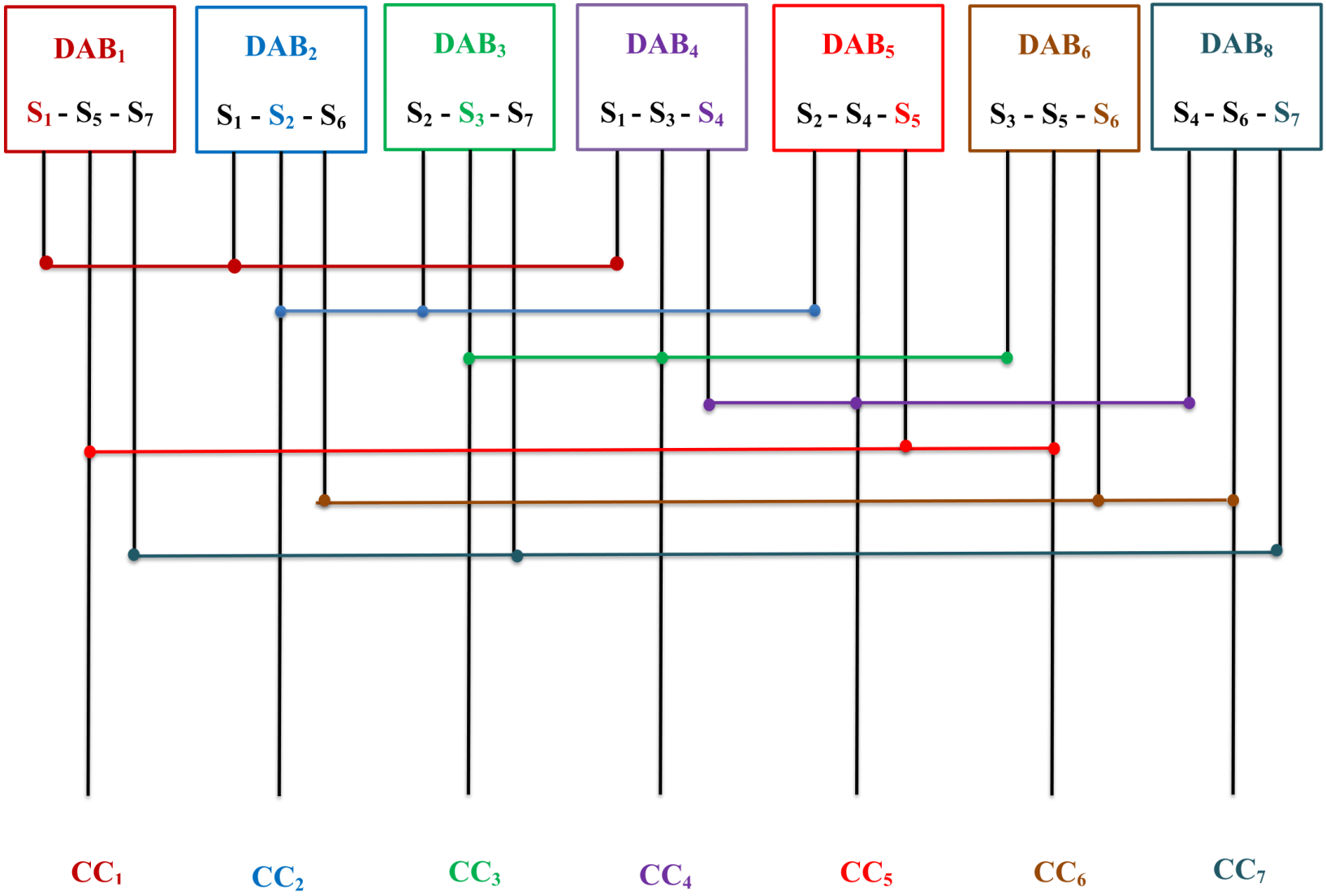
NORTHROP GRUMMAN

- 7 systems $\{S_1, S_2, S_3, S_4, S_5, S_6, S_7\}$
- 7 Distributed Autonomous Blocks (DABs) each with 3-system subset
- Each system appears in 3 DABs
- Each pair of systems appear in 1 DAB (Say, S_1 and S_5)

$$\boldsymbol{DAB_1} = \{\boldsymbol{S_1}, \boldsymbol{S_5}, S_7\}, DAB_2 = \{S_1, S_2, S_6\},$$
$$DAB_3 = \{S_2, S_3, S_7\}, DAB_4 = \{S_1, S_3, S_4\},$$
$$DAB_5 = \{S_2, S_4, S_5\}, DAB_6 = \{S_3, S_5, S_6\},$$
$$DAB_7 = \{S_4, S_6, S_7\}.$$

(7, 7, 3, 3, 1)-configuration

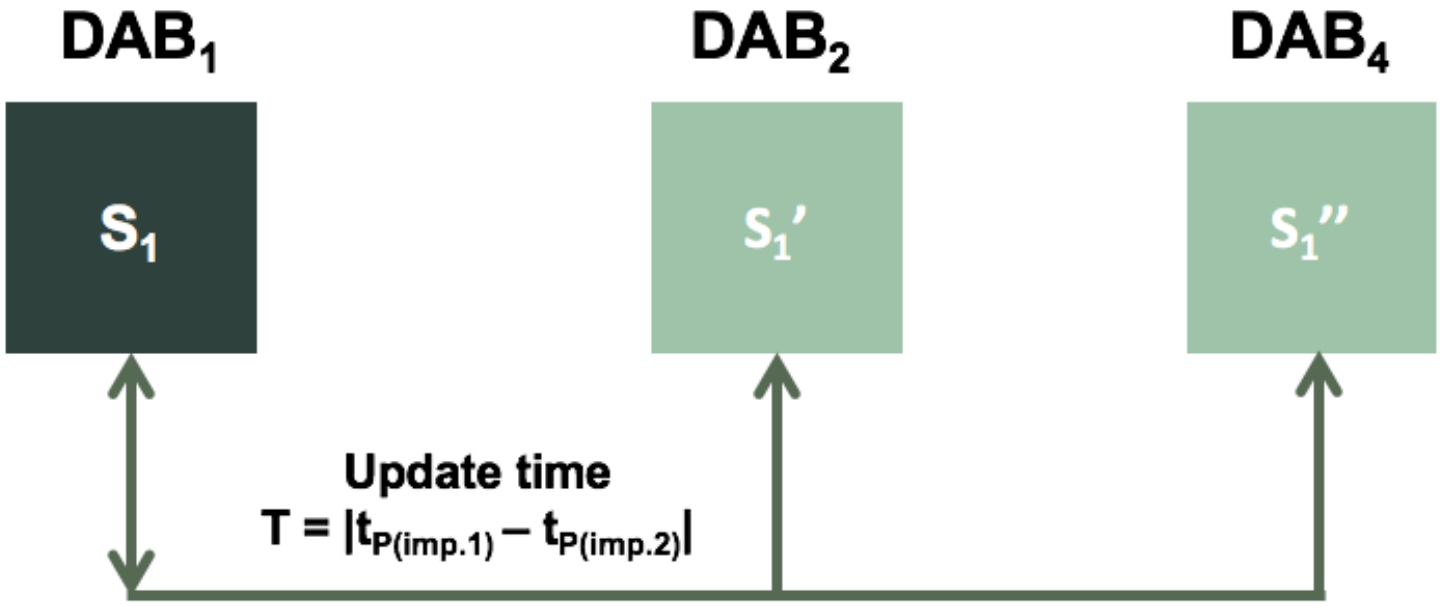
NORTHROP GRUMMAN



(7,7,3,3,1)-configuration's Functionality

NORTHROP GRUMMAN

- Each primary module periodically updates its replicas in corresponding distributed block connected by communication links (CC).



- Update the interval dynamically through learning models with Bayesian learning by continuously updating the prior.

(7,7,3,3,1)-configuration's Functionality

NORTHROP GRUMMAN

- Update time is defined as

  P_I(importance (I) | operational context (C)) = $\frac{P(C|I)P(I)}{P(C)}$

  Update interval T = | $t^1_{P(I)} - t^2_{P(I)}$ |

- When any system in any primary module's DAB acts in anomalous fashion, that system can be
  - Replaced with one of the replicas that can be selected in round robin fashion.
  - Anomalous module will be set for self-healing or repair by external source

Our Implementation and Deliverables

NORTHROP GRUMMAN

- The prototype is built with FAYE framework[1] with Node.js.
- It is a server-client framework where servers act as primary modules and clients as replicated system.
- Replica updates are done through a combinatorial design simulator[2].
- Combinatorial simulator is loaded with finite processes to compare the updates and processing time compared to a regular or sequential processing.

[1]https://faye.jcoglan.com/node.html

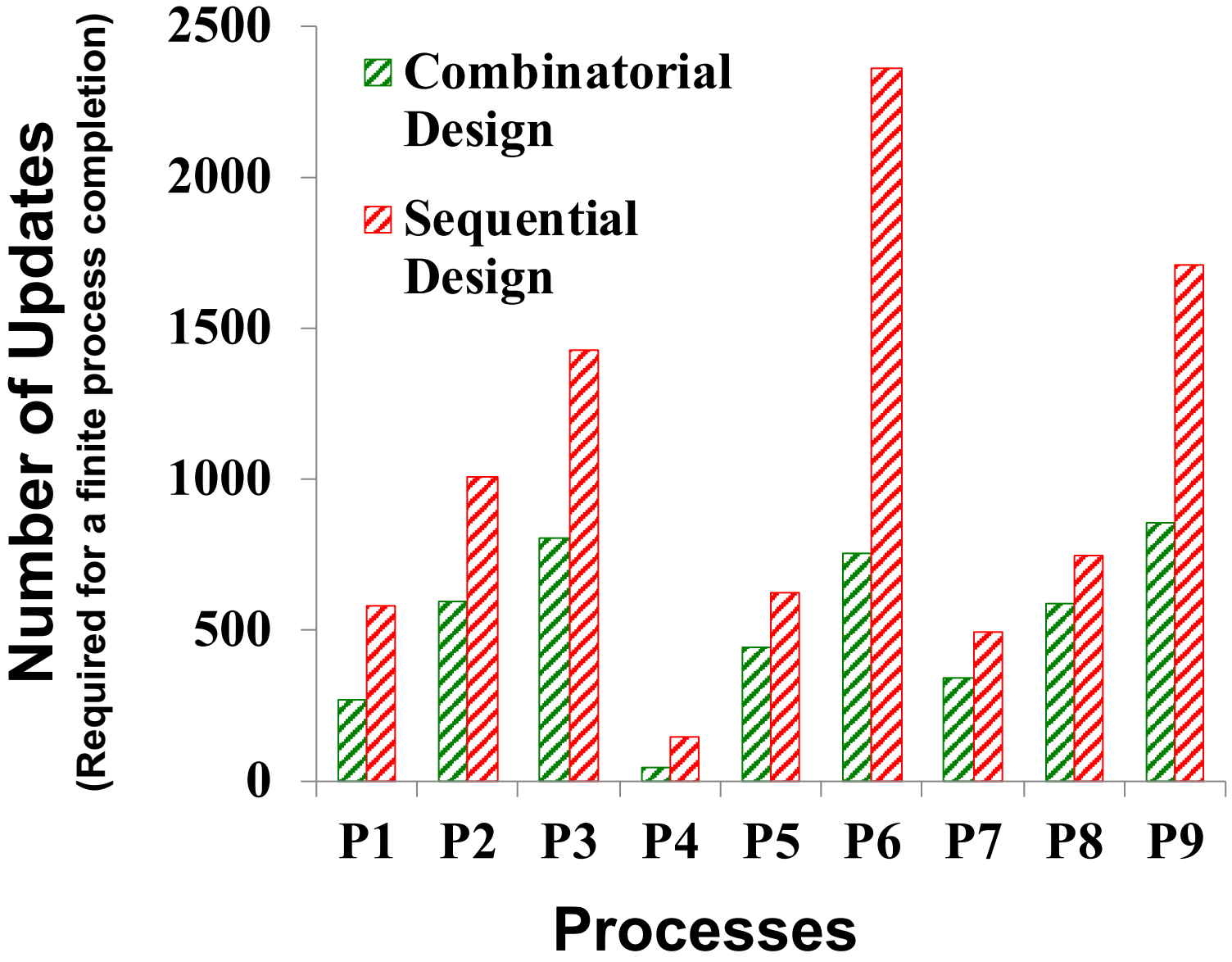
[2]https://goo.gl/pgVHdk

Our Implementation and Deliverables

NORTHROP GRUMMAN

- **Deliverables:**
  - Autonomous replica replacement prototype
  - Source code: Node.js implementation, Bayesian model, simulation software developed for combinatorial design, and Data used for simulation. Link: https://goo.gl/M4rXCN
  - Documentation: Demo video and User manual for running the prototype.

Results of Measurements



Results of Measurements

NORTHROP GRUMMAN

Process	Speed Up of Replica Scheme (Compared to regular sequential design)
FIBSEARCH	1.3
DOUBLE MULT	1.4
FIBB	1.5
SEARCH	1.8
COPY	1.8
SCALAR	2
SUM	2.1
PRINT	3
MOVEMENT	3.1

Advantage of the Design

NORTHROP GRUMMAN

- Since block sizes are equal the efficiency of the system is balanced.
  - It provides a stable and reliable system [1].
  - The design provides scalability with thousands of systems with simplistic communication links.

- The design is scalable since the number of replicas can be decreased depending on the failure rate of the whole system. (Note: Replicas are C – 1).

$$\text{No. of Replicas} \propto \frac{1}{\text{Number of Systems (N)}|\ (\text{Failure Rate} < \text{C})}$$

Advantage of the Design

NORTHROP GRUMMAN

- Updates to the replicas happen independently and simultaneously without interfering with others
- Replicas can aid other functionalities with their updates.
- All the systems in the DAB are connected which aid a faster communication.
  - For example, if S3 needs an update from S1 and S4, it can instantly communicate with them since they are in DB 4. This type of instant communication can aid parallel progressing of any process.

Combinatorial Balanced-block Structure: Other possibilities

NORTHROP GRUMMAN

- There are other configurations such as
  - M = 7, 13, 21, 31, 55, 73, 91, 132
  - N = 7, 13, 21, 31, 55, 73, 91, 132
  - C = 3, 4, 5, 6, 8, 9, 10, 12
  - R = 3, 4, 5, 6, 8, 9, 10, 12
  - Δ = 1, 1, 1, 1, 1, 1, 1, 1

Benefits to NGC IRADs and Customers

NORTHROP GRUMMAN

- The recovery and replica replacement mechanism contributes towards self-healing: Automated Enterprise Monitoring and Recovery (AEMR) IRAD.

- Distributed, simultaneous, and independent communication contributes to Distributed Data Processing IRAD.

THE VALUE OF PERFORMANCE.
NORTHROP GRUMMAN

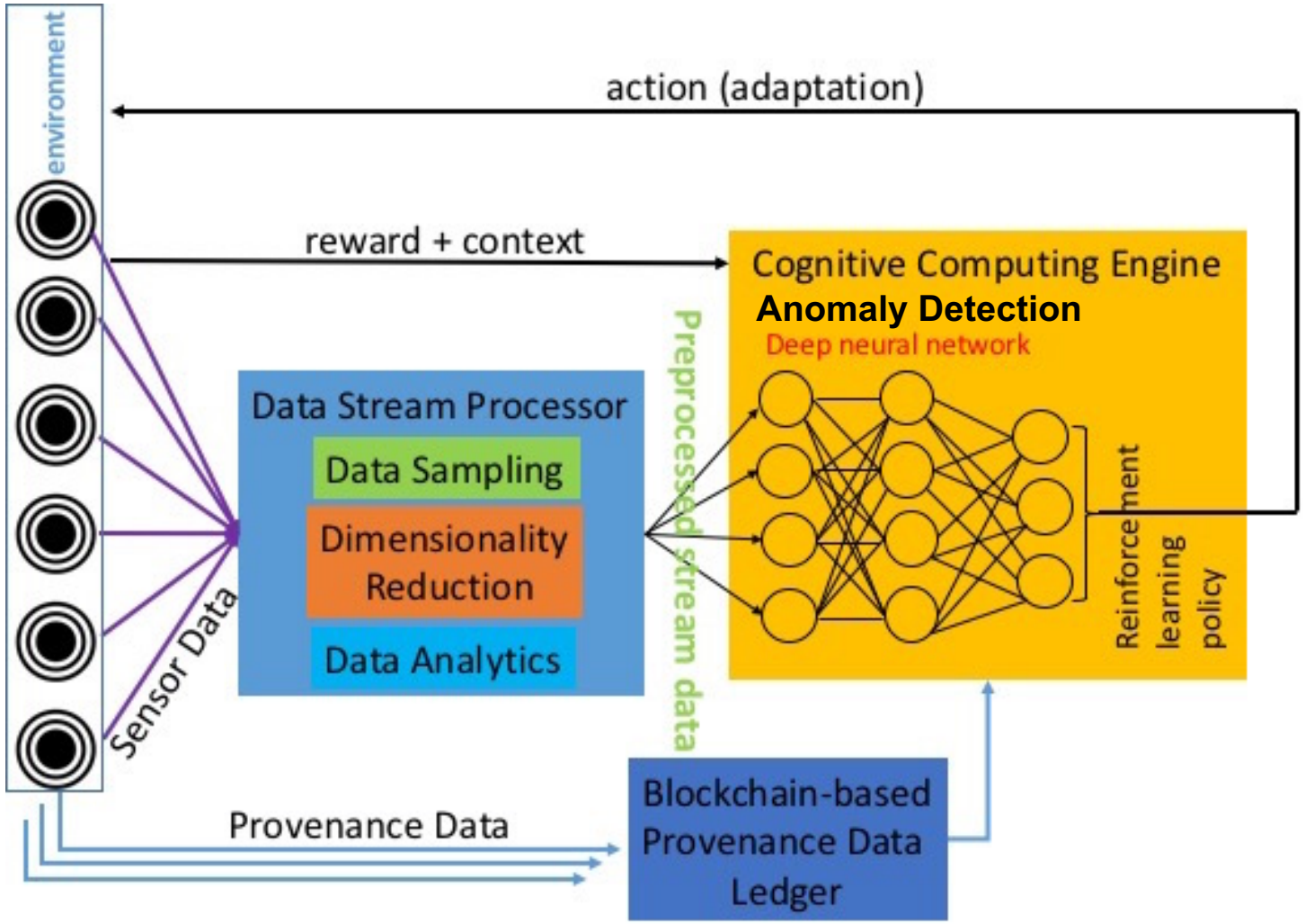
Trust

Blockhub: A Blockchain-Based Distribution of Datasets in IAS

CYBERSECURITY RESEARCH Consortium

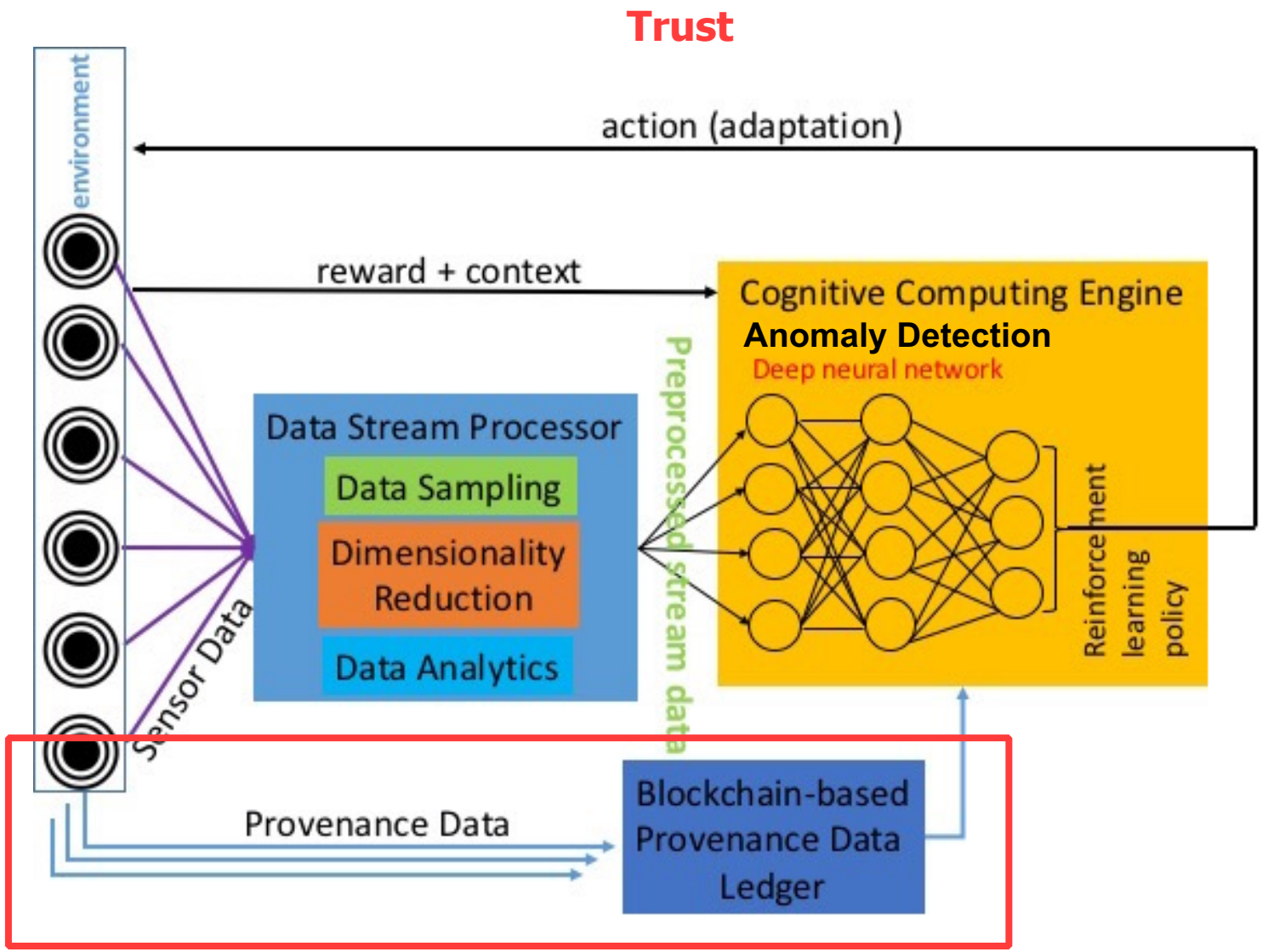
Comprehensive Architecture of IAS

NORTHROP GRUMMAN



Comprehensive Architecture of IAS

NORTHROP GRUMMAN



Problem Statement

NORTHROP GRUMMAN

- Provide trust (integrity, confidentiality, verifiability) to provenance data in IAS
  - Interactions between services are logged
  - Log records can not be corrupted
- Provide trust for network participants in IAS

  (a) ensure data confidentiality

  (b) ensure data integrity
- Provide privacy-preserving data exchange in IAS

Blockchain technology deployment challenges

NORTHROP GRUMMAN

- Performance:
  (a) Transaction latency on IBM Hyperledger Fabric blockchain platform (ver 1.0.x) is greater than 6 seconds
  (b) Performance overhead on log verification phase when the chain has large in size (a great amount of blocks) => verifiers have to compute too many hash functions

Solution:

(a) modify transaction verification process at “Endorsers”
(b) Use depth-robust graphs to store blockchain (in collaboration with Prof. Blocki, Purdue) which can validate transaction without checking every block in the chain

Blockchain technology deployment challenges

NORTHROP GRUMMAN

- Fine-grained role- and attribute-based access control with data leakage detection capabilities

Solution:

Integrate WAXEDPRUNE project into blockchain-based framework to provide:

(a) role-based access control

(b) detection/prevention of data leakages made by insiders

(c) attribute-based access control, with attributes including:

(c1) trust level of network nodes

(c2) context (e.g. normal vs. emergency)

(c3) authentication method (e.g. password-based vs. fingerprint)

(c4) cryptographic capabilities of network node

Blockchain technology deployment challenges

NORTHROP GRUMMAN

- Failure Recovery (proposed by Steve Seaberg, NGC)
  - Need to maintain consistency in mobile environment with intermittent connectivity
  - Need quantification of performance parameters after a varying period of connectivity breakdown
  - Need to determine how much bandwidth and resources are needed to make network nodes consistent (or current)

Blockchain technology deployment challenges

NORTHROP GRUMMAN

- *Access Control (Read):* revoked access to data in a blockchain can be bypassed as follows:
  - Attacker holding a copy of a blockchain can use a modified client to just ignore the revocation block
  - Attacker can replay old blocks against an empty blockchain and stop before revocation block is appended

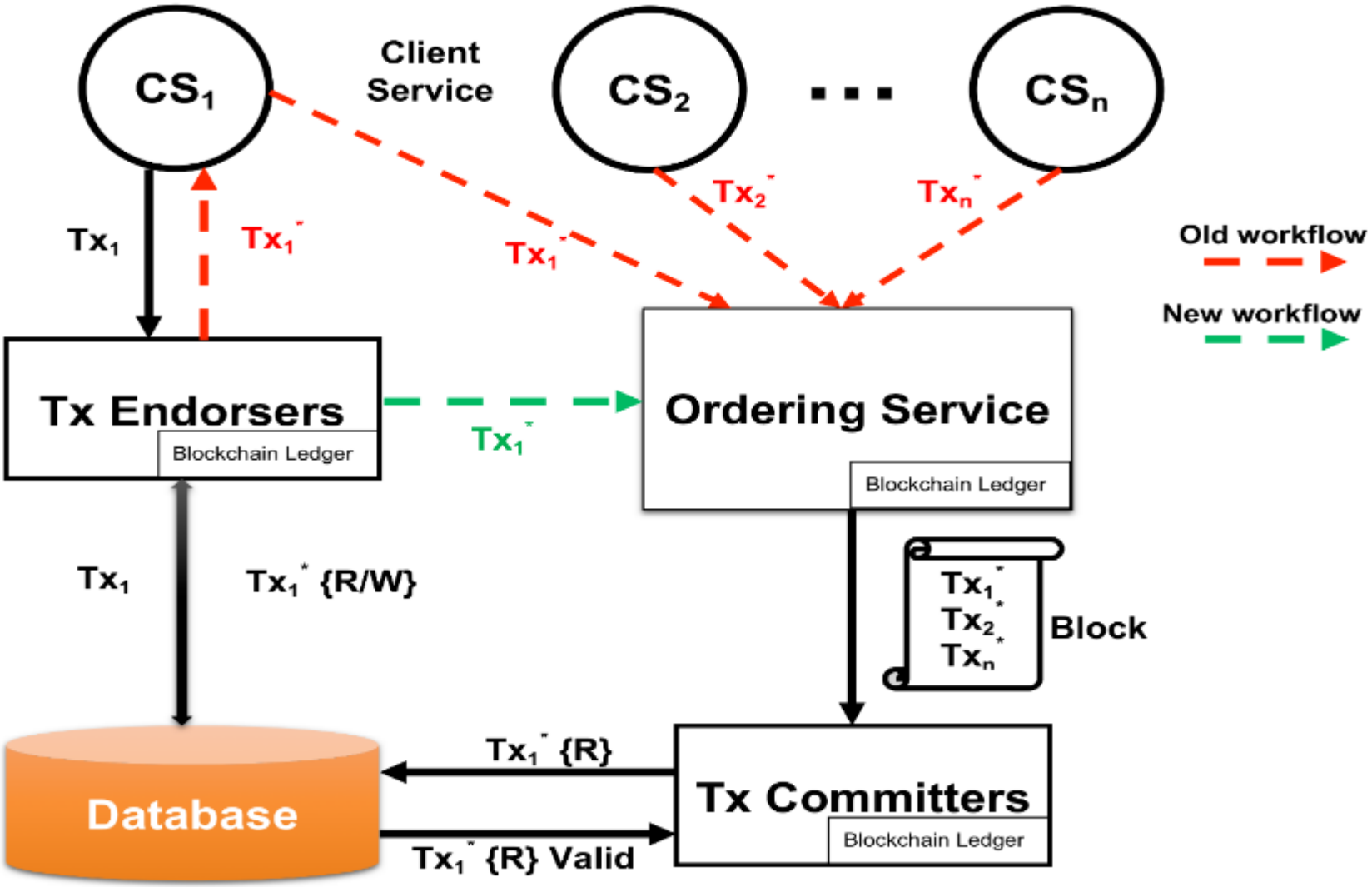
Blockchain technology deployment challenges

NORTHROP GRUMMAN

- Provide secure User Interface to create Smart Contracts (Chaincodes)
  - Idea proposed by Ashish Kundu (IBM)
  - GO language is currently used to write chaincode
  - How to make chaincode writing easier for the user and reduce attack surface

Solution for Permissioned Systems

NORTHROP GRUMMAN



Research Approach

NORTHROP GRUMMAN

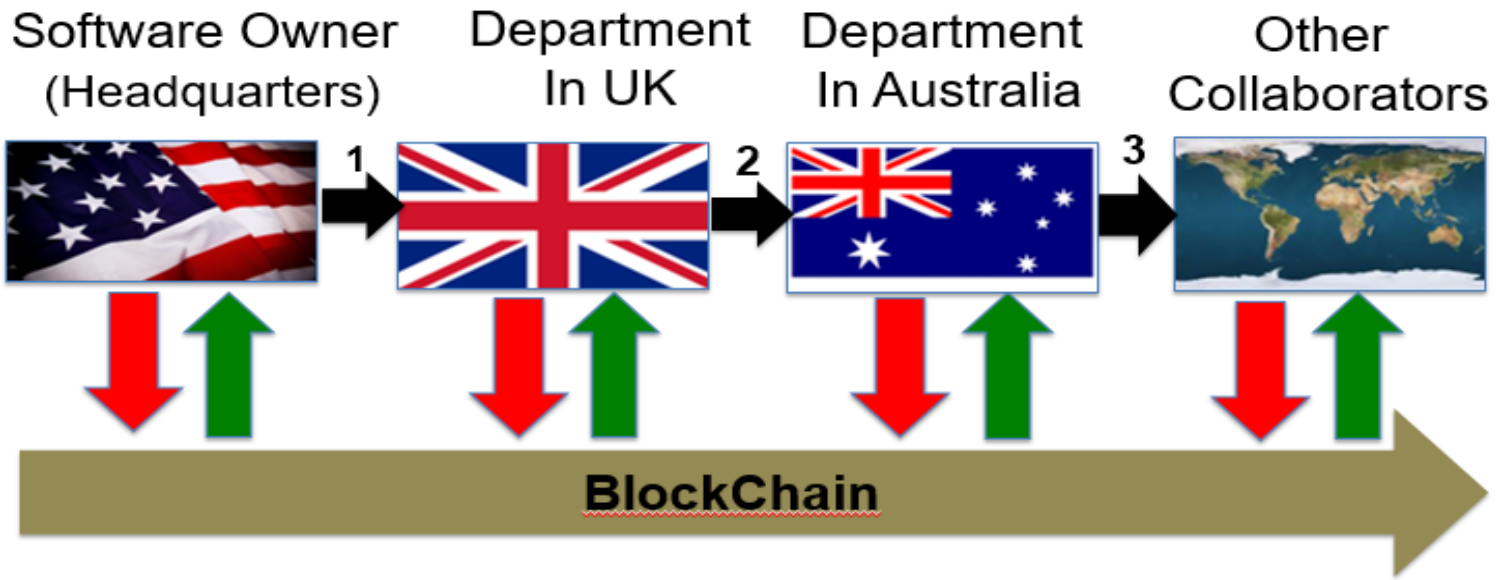
- Idea: protect against dDOS attacks when invalid transactions flood the network and don't let valid transactions to be processed
  - Remove client service's communication with the ordering service and extra-communications with Endorsers
  - Endorsers directly send validated transaction to the Ordering Service
  - Ordering Service delivers transaction to the Committers

Blockhub: blockchain-platform for IAS

NORTHROP GRUMMAN

Blockchain-based Software Distribution (WAXEDPRUNE extension)

- Blockchain-based technology ensures integrity of provenance data
- *BlockHub* provides secure cross-domain software distribution



Blockhub: blockchain-platform for IAS

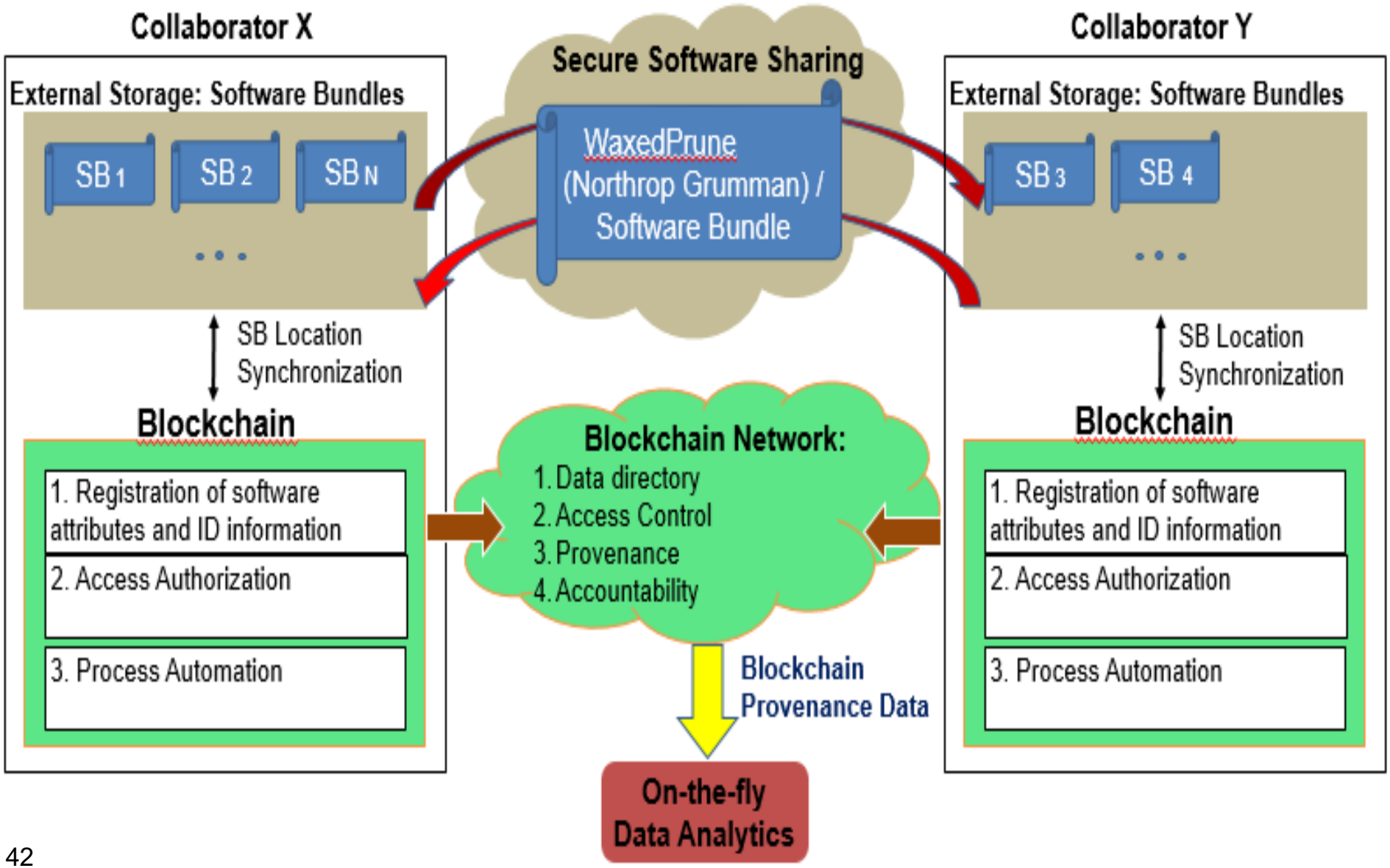
NORTHROP GRUMMAN

BlockHub can be used for: (proposed by Robert Pike, NGC)

1. Tracking and control of software components that are shared across multiple security domains.
2. Automating the export auditing and tracking processes.
3. Cross-domain dissemination of encrypted software modules using role- and attribute-based access control.
4. Licensing provenance of deployed software components.
5. Enabling software supply chain that is tamper resistant.
6. Software spillage remediation.

Blockhub: blockchain-platform for IAS

NORTHROP GRUMMAN

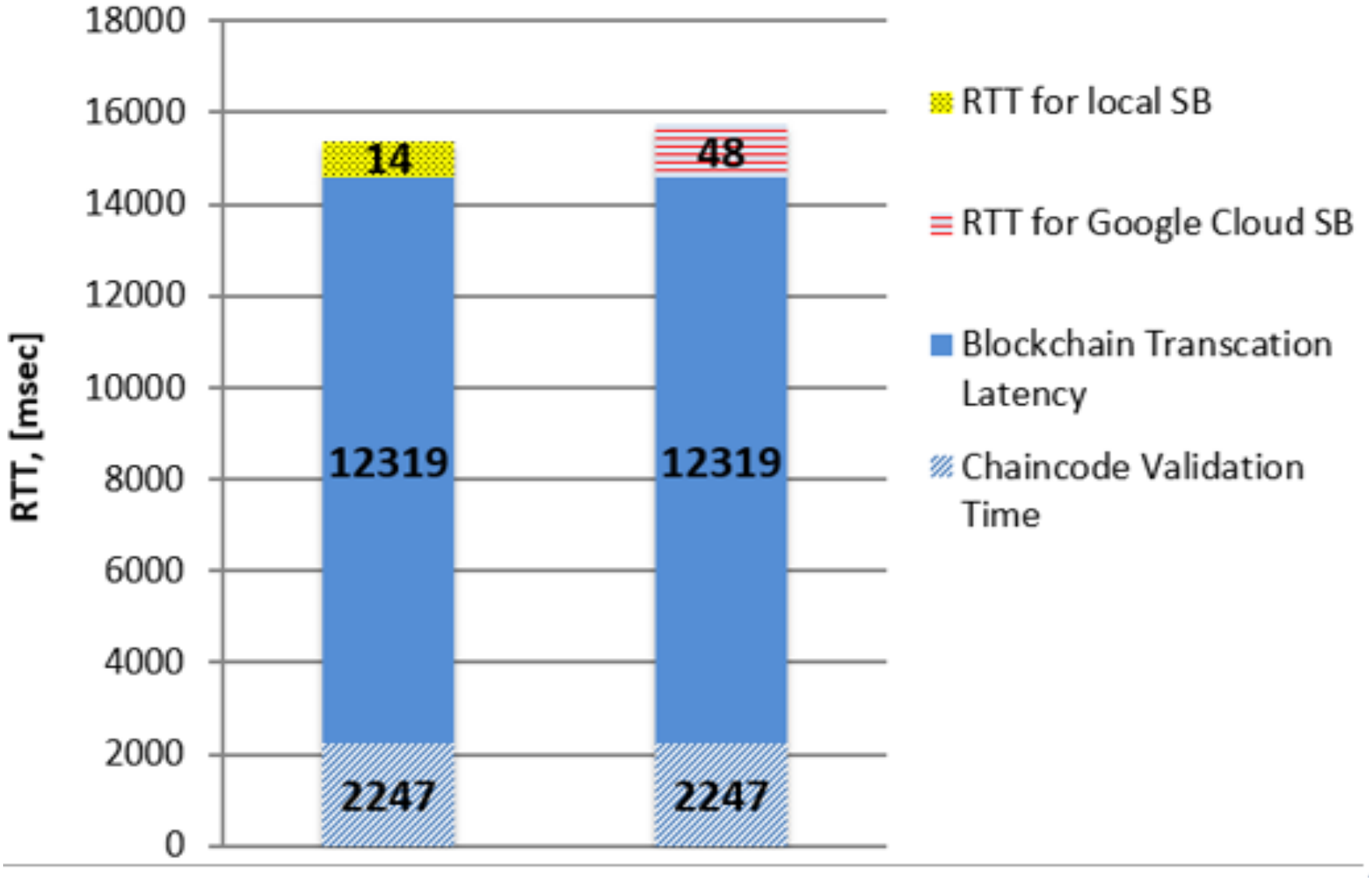


Blockhub Architecture

NORTHROP GRUMMAN

- X and Y share software via *smart contracts* running in blockchain network
- Every request is logged in the blockchain's distributed ledger
- Software stored in form of Software Bundles (SB) that contain:
  - *Encrypted Software Modules (source code or executables)*
  - *Access Control Policies*
  - *Policy Enforcement Engine (Virtual Machine)*
- Software is transferred if authorization has been granted by both smart contract and policy enforcement engine of the SB
- Any transaction, i.e. software access/update can be verified any time in the future
- Storing blockchain in form of *depth-robust graph* (proposed by Prof. Blocki, Purdue Univ.) reduces transaction validation time
- There is one depth-robust graph per SB and it gets updated each time a transaction occurs

Evaluation


18000
16000
14000
12000
10000
8000
6000
4000
2000
0
RTT, [msec]
14
12319
2247
48
12319
2247
RTT for local SB
RTT for Google Cloud SB
Blockchain Transcation Latency
Chaincode Validation Time

Implementation and Deliverables

NORTHROP GRUMMAN

- **Deliverables:**

  'Blockhub' prototype for secure blockchain-based data distribution

  - Source code: https://github.com/Denis-Ulybysh/Waxedprune2018
    * codebase is taken from open-source "Marbles" project https://github.com/IBM-Blockchain/marbles/tree/v4.0

  - Demo video

THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN

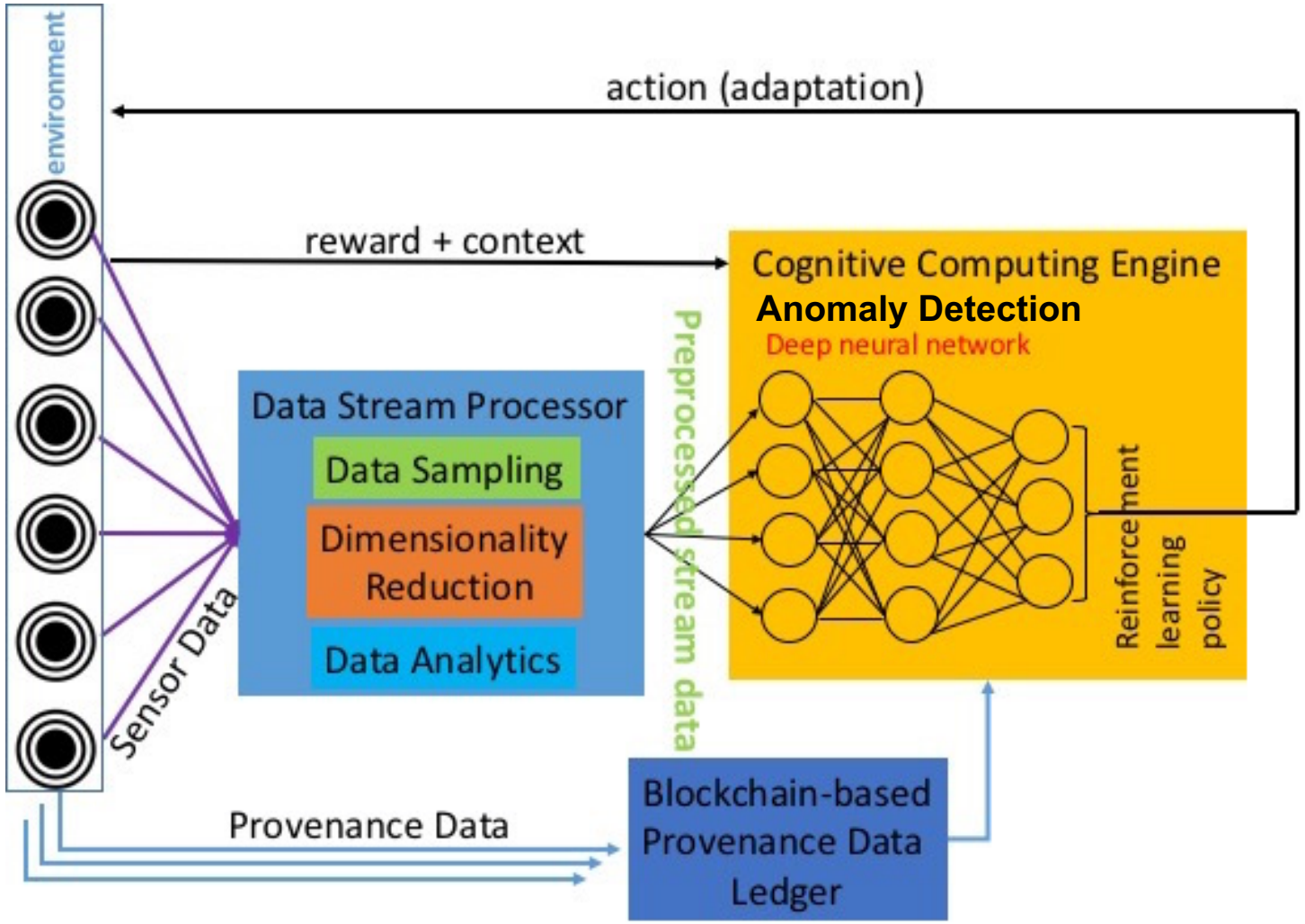
Cognitive Autonomy / Knowledge Discovery

A Deep Learning Based Anomaly Detection Solution

CYBERSECURITY RESEARCH Consortium

Comprehensive Architecture of IAS

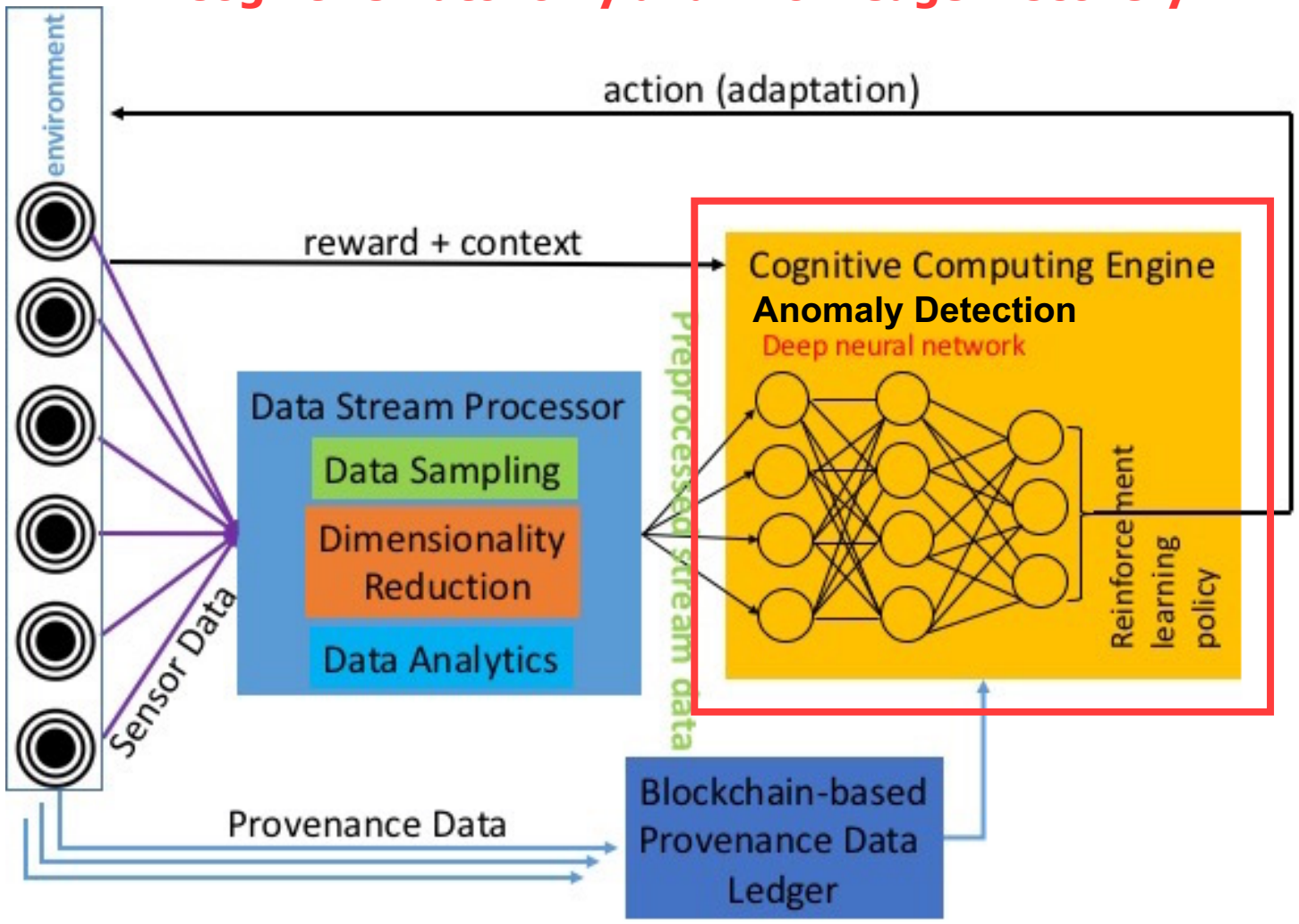
NORTHROP GRUMMAN



Comprehensive Architecture of IAS

NORTHROP GRUMMAN

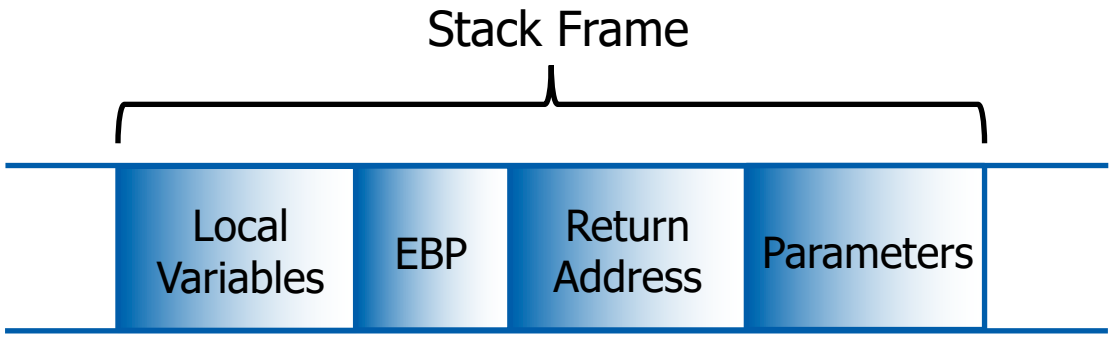
Cognitive Autonomy and Knowledge Discovery



Problem Statement

NORTHROP GRUMMAN

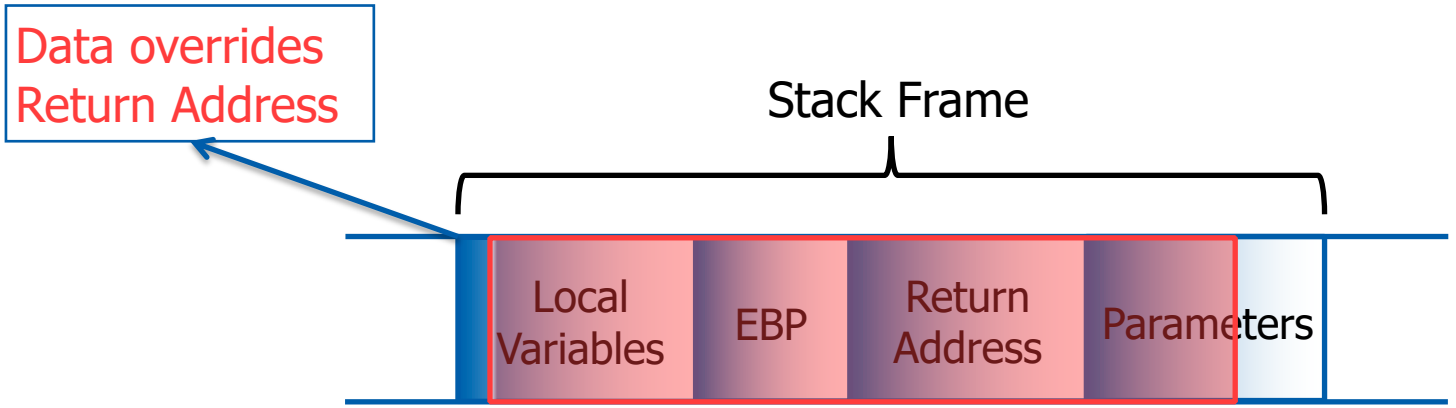
- **Intelligent Autonomous Systems (IAS)** adapt to meet the mission objectives without disrupting the ongoing critical processes by incremental learning (**reflexivity**).
- Programs store Return Addresses (control flow) along with data in the stack.
- **Control-hijacking** attacks execute arbitrary code on the target IAS program by hijacking its control flow.



Problem Statement

NORTHROP GRUMMAN

- **Intelligent Autonomous Systems (IAS)** adapt to meet the mission objectives without disrupting the ongoing critical processes by incremental learning (**reflexivity**).
- Programs store Return Addresses (control flow) along with data in the stack.
- **Control-hijacking** attacks execute arbitrary code on the target IAS program by hijacking its control flow.



Problem Statement

NORTHROP GRUMMAN

- Advanced modern exploits characterize by their sophistication in stealthy attacks.
- Code-reuse attacks such as return-oriented programming (ROP) and memory disclosures allow attackers executing malicious instruction sequences on victim systems without injecting external code.
- Control Flow Integrity (CFI) is required.
- **Research Question:** Can we developed a Deep Learning based anomaly detection technique that probabilistically models program control flows for behavioral reasoning and system monitoring?

Research Approach

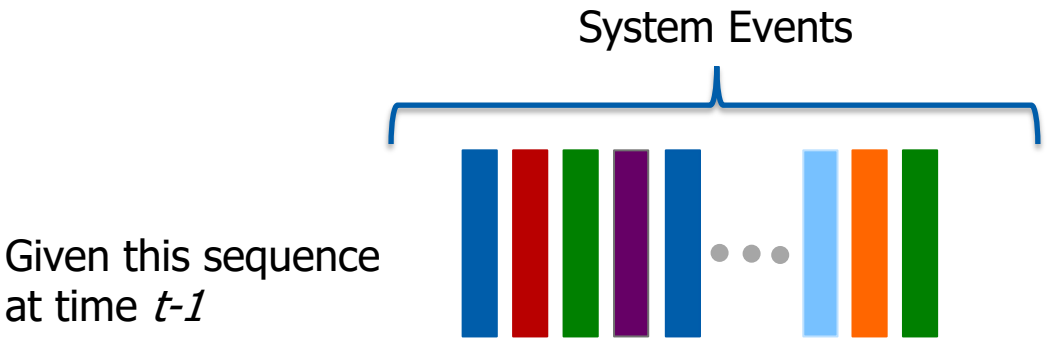
NORTHROP GRUMMAN

- An event e_i *is defined* defined as a function call (**system or library call**) in the execution trace of a program.
- Use Deep Learning to answer **the binary classification problem** of given a sequence of function calls (or system events) $e_1e_2e_3…e_k$ **whether or not the sequence should occur**?

Research Approach

NORTHROP GRUMMAN

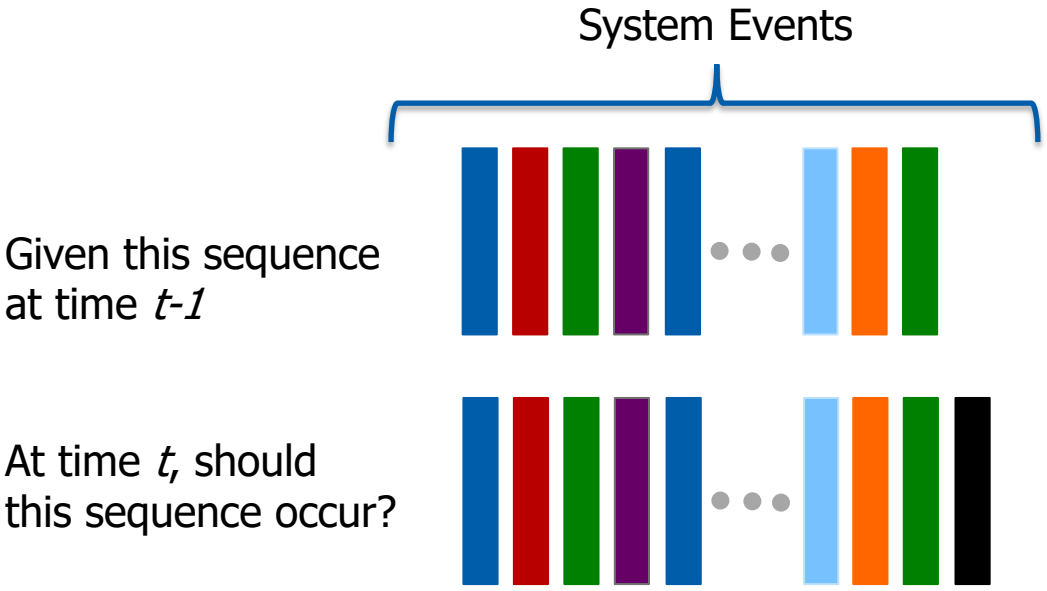
- An event e_i *is defined* defined as a function call (**system or library call**) in the execution trace of a program.
- Use Deep Learning to answer **the binary classification problem** of given a sequence of function calls (or system events) $e_1e_2e_3\ldots e_k$ **whether or not the sequence should occur**?



Research Approach

NORTHROP GRUMMAN

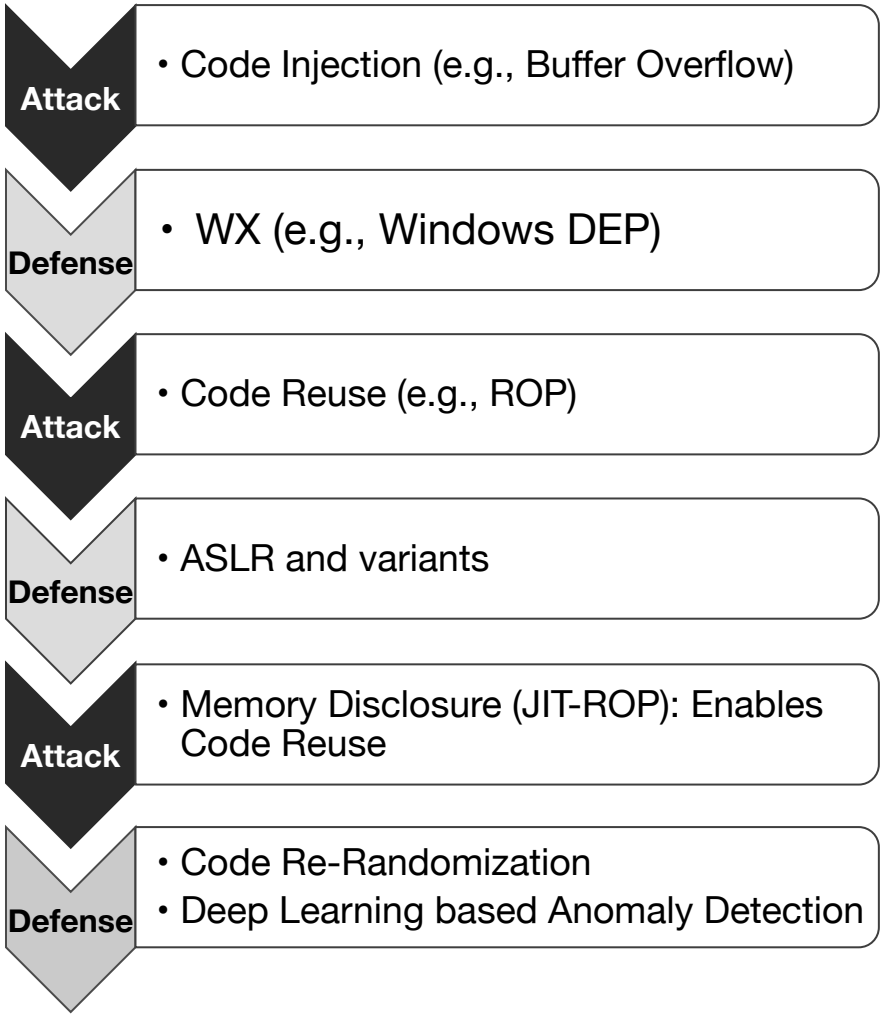
- An event e_i *is defined* defined as a function call (**system or library call**) in the execution trace of a program.
- Use Deep Learning to answer **the binary classification problem** of given a sequence of function calls (or system events) $e_1e_2e_3…e_k$ **whether or not the sequence should occur**?



Deep Learning against code reuse attacks

NORTHROP GRUMMAN

The eternal war



❑ **Invalid and abnormal control flow of a program:**

✓ Code Injection
✓ Code Reuse
✓ Memory Disclosure

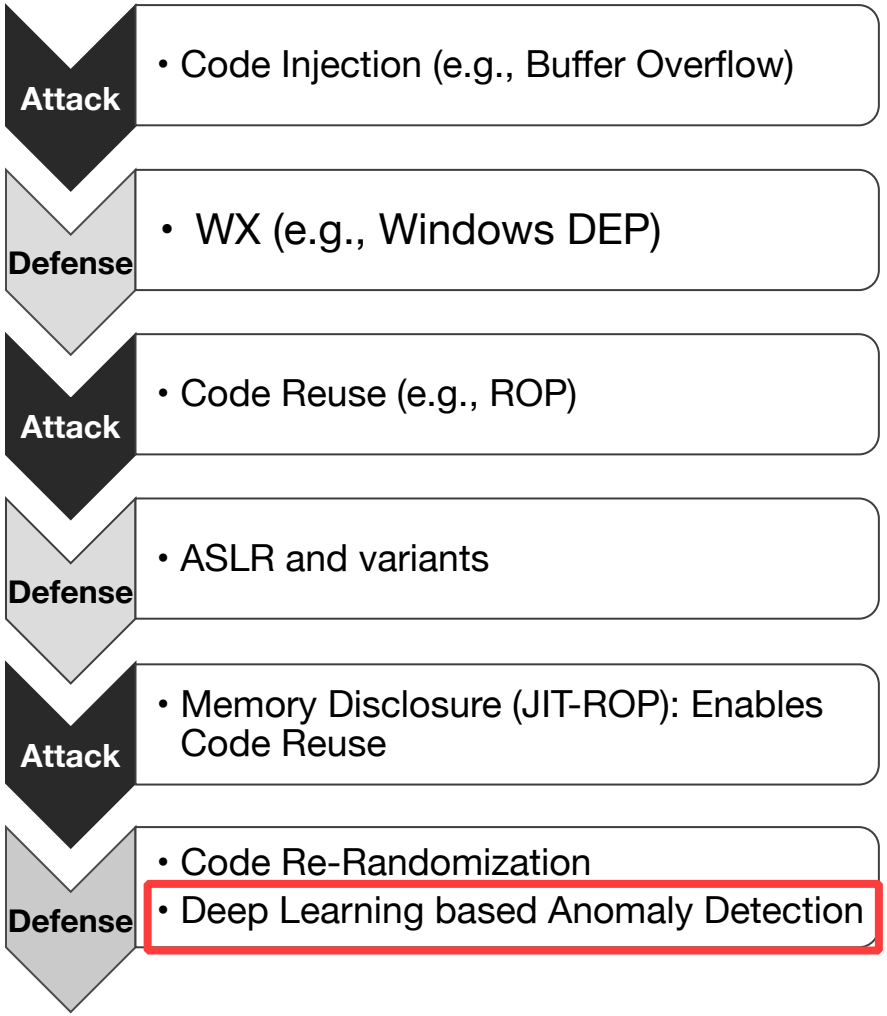
❑ **Can be caused by:**

✓ Human error (e.g., unauthorized use or operation of the program
✓ Software flaws (e.g., buffer overflow vulnerabilities)
✓ Attacks by remote attackers
✓ Malicious insiders (e.g., through drive-by downloads)

Deep Learning against code reuse attacks

NORTHROP GRUMMAN

The eternal war: DL against code reuse attacks



- ❑ **Invalid and abnormal control flow of a program:**
  - ✓ Code Injection
  - ✓ Code Reuse
  - ✓ Memory Disclosure
- ❑ **Can be caused by:**
  - ✓ Human error (e.g., unauthorized use or operation of the program
  - ✓ Software flaws (e.g., buffer overflow vulnerabilities)
  - ✓ Attacks by remote attackers
  - ✓ Malicious insiders (e.g., through drive-by downloads)

Other Learning Based Approaches

NORTHROP GRUMMAN

Classification Model Based Approaches:

Attack	*Feature 1*	*Feature 2*	*…*	*Feature N*
+			…	
-			…	
+			…	
-			….	

- Datasets include known attacks only

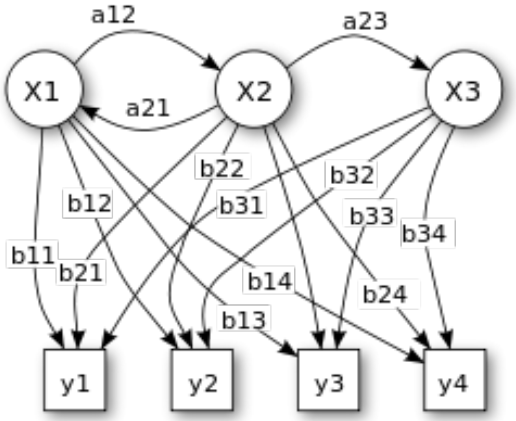
Limitation:

- No resilient to zero-day exploits (Signature Based)
- Difficult to train (representative datasets of the attacks are required)

Other Learning Based Approaches

NORTHROP GRUMMAN

Hidden Markov Model (HMM) Based Approaches:



- Number of hidden states is equal to the number of system events
- $P(Y) = P(Y|X)\ P(X)$

Limitation:

- **No memory nor long dependencies**: Observation at time *t* depends on observation at time *t-1*

Our Deep Learning Based Approach

NORTHROP GRUMMAN

- For a given program, a code coverage is conducted to obtain all the possible execution traces.
- An event e_i *is defined* defined as a function call (**system or library call**) in the execution trace of a program.
- Each possible system event (function calls) is uniquely identified as they will form the vocabulary of system events.
- The Deep Learning model (neural network) is trained with the obtained sequences of events.
- The model is based on Recurrent Neural Networks: Long-Short Term Memory (**LSTM**) and Gated Recurrent Units (**GRU**.)

Our Deep Learning Based Approach

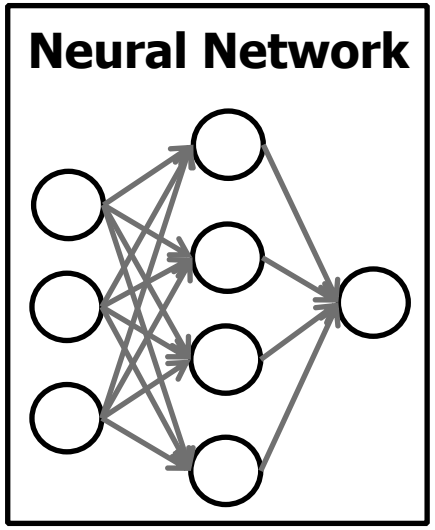
NORTHROP GRUMMAN

- After training, given a sequence of events as input, the neural network produces as output an array of probabilities, one for each of the possible events in the system.
- At any time *t* each possible event (system call or library call) in the system is assigned a probability estimated with respect to the sequences of events **observed until** time *t-1*.
- At classification time *t*, the decision is made with respect to a pre-defined threshold of the top-*k* most likely events.

Our Deep Learning Based Approach

NORTHROP GRUMMAN

Set of all system events

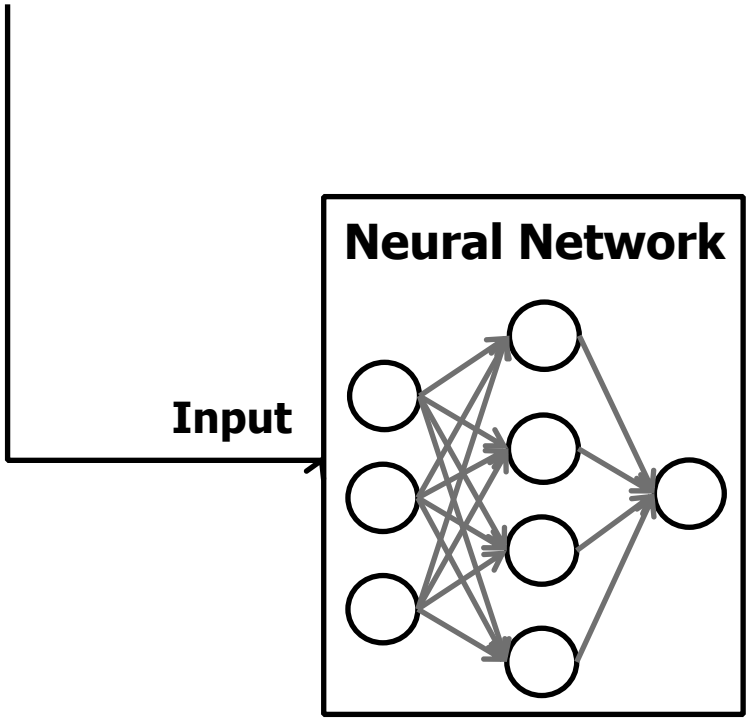


Our Deep Learning Based Approach

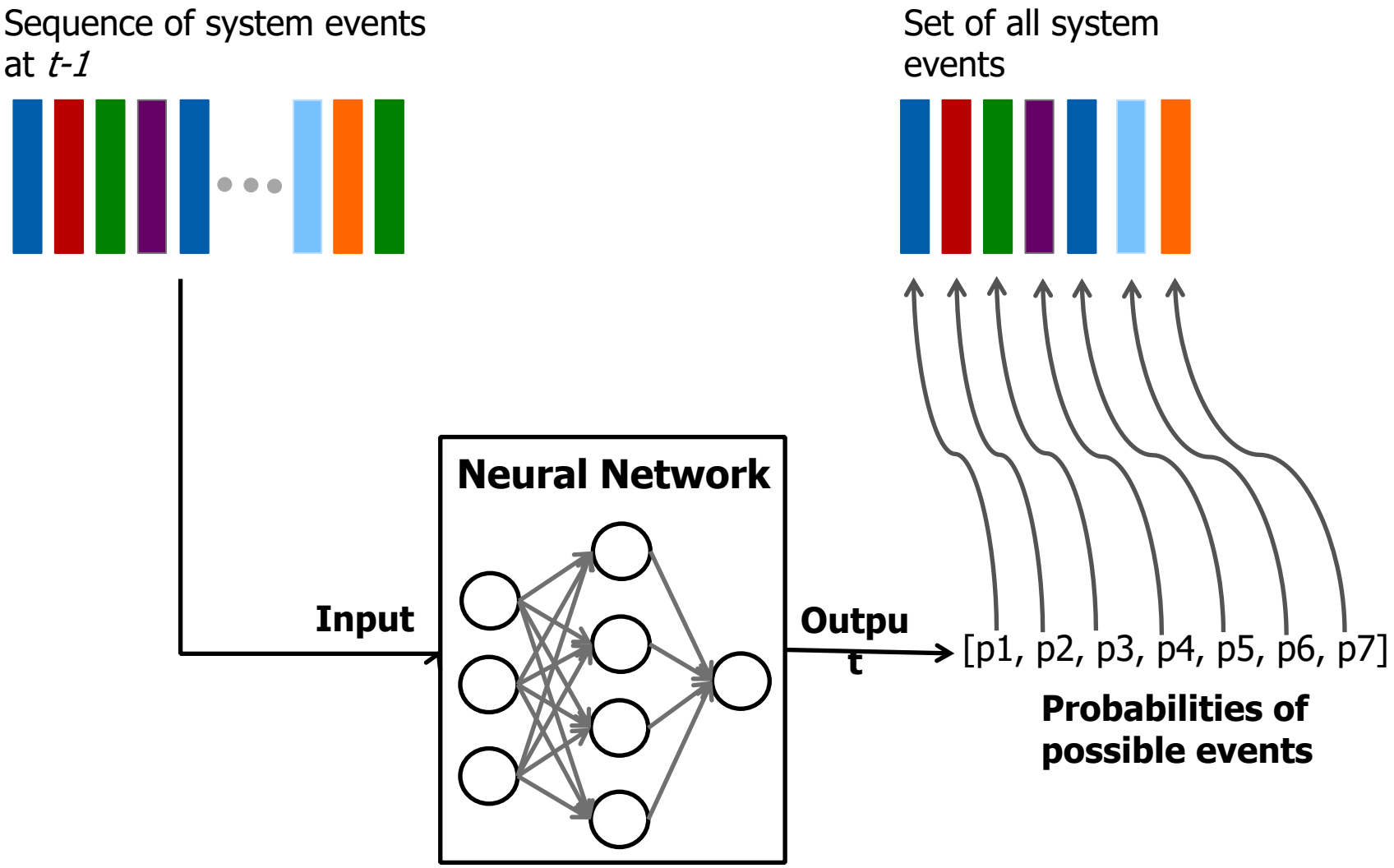
NORTHROP GRUMMAN

Sequence of system events at *t-1*

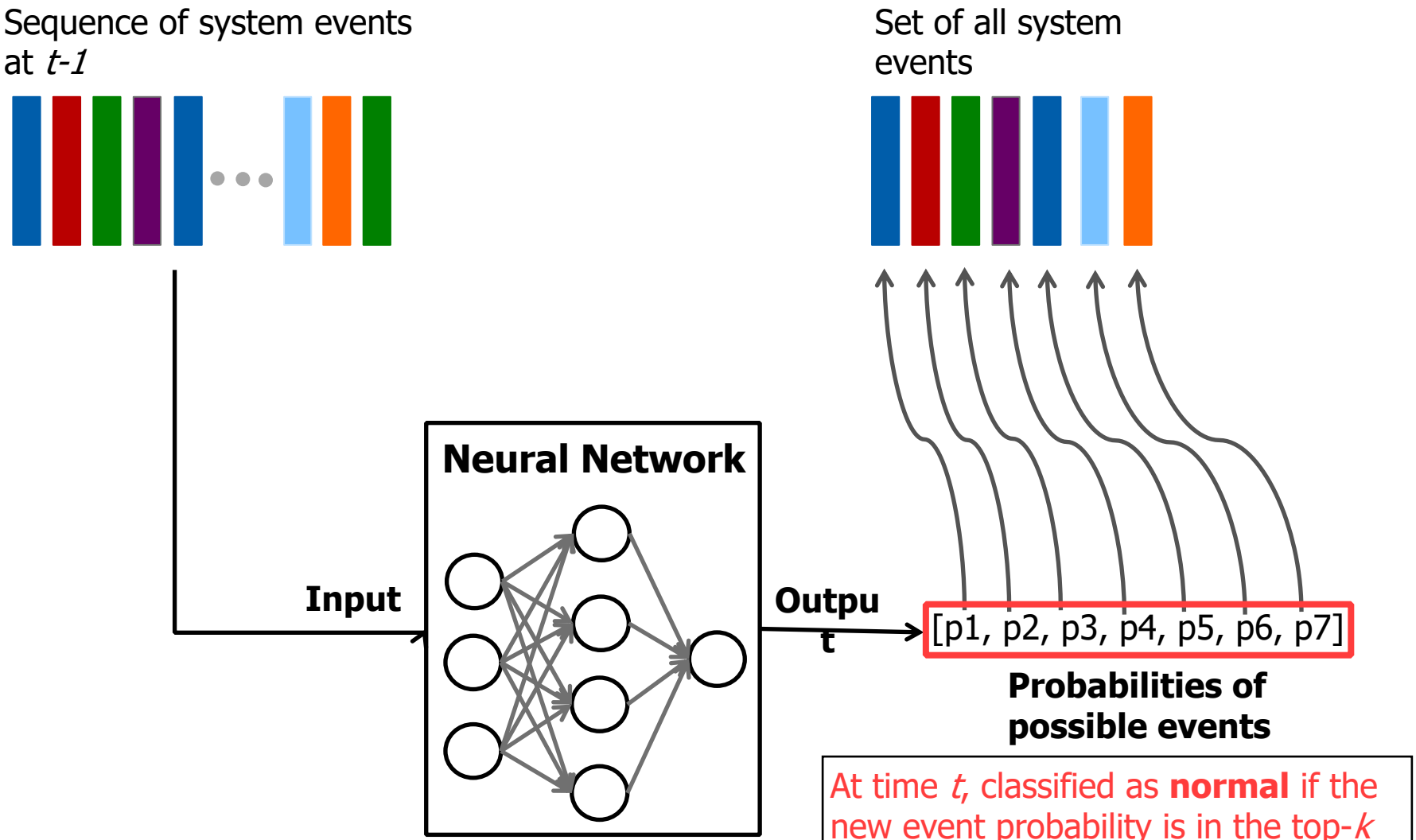
Set of all system events



Our Deep Learning Based Approach



Our Deep Learning Based Approach



Our Deep Learning Based Approach

NORTHROP GRUMMAN

Input: Sequence of events in the system
Output: normal or anomalous

- **Step 1:** Define a finite set E of events $e_1, e_2, ..., e_N$ in the system. Events occur in a time-series fashion.
- **Step 2:** At time $t-1$, given an observed series of events $\{e_i{}^1, e_i{}^2, ..., e_i{}^{t-1}\}$ (with $i = 1, 2, ...orN$) find the set K of the top k events to occur in time t.
- **Step 3:** At time t, the sequence $\{e_i{}^1, e_i{}^2, ..., e_i{}^{t-1}, e_i{}^t\}$ is non-anomalous if $e_i{}^t \in K$, otherwise anomalous.

Algorithm 1: Anomaly detection algorithm

Benefits of Our Deep Learning Approach

NORTHROP GRUMMAN

Properties of Recurrent Neural Networks (RNN)

Stateful Representation	**Memory**
Long-Term Dependency	**Variable Length Sequence**

- **Stateful Representation**: Maintenance of current state.
- **Memory**: Update state based on previous events.
- **Long-Term Dependency**: Track previous dependencies for long terms.
- **Variable Length Sequence**: Functional with non-fixed-length sequences.

Benefits of Our Deep Learning Approach

NORTHROP GRUMMAN

- **Training datasets are full available**: Dynamic code behaviors are learned by training the model with non-malicious program traces in isolation.
- **Flow sensitive anomaly detection**: Given the execution paths P_1, P_2 and P_3, our technique captures the occurrence probabilities of each of their events; vital for high-precision anomaly detection.
- **Higher Power of Representation**: Context attributes such as caller functions can be added (e.g., function call read@faa is different from function call read@foo)
- **Higher granularity of the execution traces**: More granular traces are used for training, which improves precision in detection.

Platforms and Prototype

NORTHROP GRUMMAN

Model Implementation

MODEL

```
# -*- coding: utf-8 -*-
"""
Deep Learning Based Anomaly Detection Model
"""
# Imports
from __future__ import print_function
import torch
from   torch.autograd import Variable
```

python

PYTORCH

theano

TensorFlow™

CUDA

GPU

- ❑ **Model**
  - ✓ Based on Recurrent Neural Networks (RNN): Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU)
- ❑ **Programming Language: Python**
  - ✓ Compatible with several numerical computing libraries suitable for the use of GPUs
  - ✓ Some examples: Pytorch, TensorFlow and Theano
- ❑ **Computing Library: PyTorch**
  - ✓ Scientific computing package
  - ✓ Replacement of Numpy to take advantage of the power of GPUs
  - ✓ Python-friendly platform for neural networks (Deep Learning)
- ❑ **Parallel Computing Platform: CUDA**
  - ✓ Programming model to use GPUs for general purpose computing

Publications

NORTHROP GRUMMAN

1. G. Mani, B. Bhargava, B. Shivakumar, J. Kobes "Incremental Learning Through Graceful Degradations in Autonomous Systems", IEEE ICCC, June 2018 (In Submission).

2. G. Mani, B. Bhargava, P. Angin, M. Villarreal-Vasquez, D. Ulybyshev, J. Kobes "Machine Learning Models to Enhance the Science of Cognitive Autonomy", IEEE ICCC, June 2018 (In Submission)

3. G. Mani, B. Bhargava"Scalable Learning Through Error-correcting Codes based Clustering in Autonomous Systems", IEEE ICCC, June 2018 (In Submission)

4. G. Mani, D. Ulybyshev, B. Bhargava, J. Kobes, P. Goyal"Autonomous Aggregate Data Analytics in Untrusted Cloud", IEEE ICCC, June 2018 (In Submission).

5. Mani, Ganapathy, Bharat Bhargava. "Graceful Degradation in Autonomous Systems Based on Combinatorial Learning Model". (In Submission).

6. D. Ulybyshev, M. Villarreal-Vasquez, B. Bhargava, G. Mani, S. Seaberg, P. Conoval, D. Steiner, J. Kobes "Blockhub: Blockchain-based Software Development System for Untrusted Environments", IEEE CLOUD 2018, (In Submission).

7. D. Ulybyshev, B. Bhargava, A. Alsalem "Secure Data Exchange and Data Leakage Detection in Untrusted Cloud", ICACCT 2018 (Accepted, in-press).