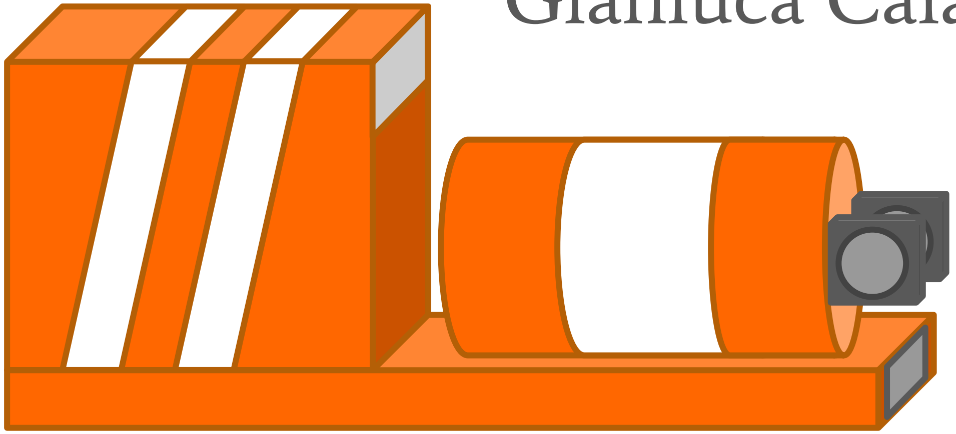
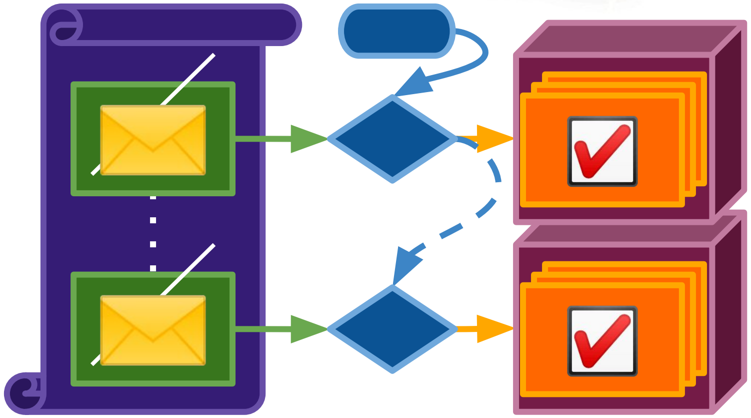
Black Block Recorder:
Immutable Black Box Logging via rosbag2 and DLTs

By: Ruffin White,
Gianluca Caiazza





Motivation

With maturing R&D, robots have propagated into many industries and economies, including manufacturing, transport, logistics, and more.

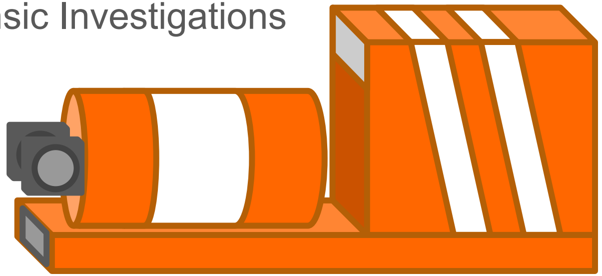
As cyber-physical systems (CPS), the risks from and threats to robots can surpass that incurred by conventional Internet of Things (IoT) devices.

Yet, are the methods used to monitor/audit such CPS devices in pace with robotic proliferation?

Motivation

Event Data Recorders (**EDRs**) in automotive, or Black Box Flight recorders in aerospace industries serve a multitude of critical directives, including:

- Dataset & Model Acquisition
- Performance Benchmarking
- Quality Control and Testing
- Continuous Health Monitoring
- Diagnostic Debugging & Triage
- System Auditing & Verification
- Digital Forensic Investigations



Motivation

With high-stake deployments come high-stake consequences, thus the extraordinary incentives evident to **secure** vs. **circumvent** EDR devices:

- Hold liable parties accountable
- Guide regulatory & legislative policy

VS.

- Expunge incriminating evidence
- Conspire to falsify historic events

Directive: to verify the *integrity*, *authenticity*, and *completeness* of robotic logs under the threat of malicious/erroneous <u>insertion</u>, or <u>omission</u>.

The five Robots

Death by robot: the new mechanised danger in our changing world

As the use of autonomous machines increases in society, so too has the chance of robot-related fatalities

NEWS

CENTS TUESDAY, DECEMBER 8, 1981

U.S.

$10 million awarded to family of plant worker killed by robot

Knight-Ridder Newspapers

DETROIT The manufacturer of a one-ton robot that killed a worker at Ford Motor Co.'s Flat Rock casting plant must pay the man's family $10 million, a Wayne County Circuit Court jury ruled this week.

The jury of three men and three women deliberated for 2⅞ hours before announcing the decision against Unit Handling Systems in a suit by the family of Robert Williams, who was killed Jan. 25, 1979. Unit Handling is a division of Litton Industries.

It is believed to be the largest personal injury award in state history.

At the time of his death, Williams, 25, of Dearborn Heights, Mich., was one of three men who operated an electronic parts-retrieval system at Ford's Flat Rock plant. The plant has since been closed.

The system, made by Unit Handling, was designed to have a robot automatically recover parts from a storage area at the plant.

On the day of his death, Williams was asked to climb into a storage rack to retrieve parts because the robot was malfunctioning at the time and not operating fast enough, according to the Williams family's attorneys.

The robot, meanwhile, continued to work silently, and a protruding segment of its arm smashed into Williams's head, killing him instantly.

The robot kept operating while Williams lay dead for about 30 minutes. His body was discovered by workers who became concerned because he was missing.

Attorneys for the family said the robot should have been equipped with devices to warn workers that it was operating.

"If they didn't want people up there when the robot was moving around, they should have installed safety devices," said Joan Lovell, one of the two attorneys representing the family. "Human beings are more important than production."

The jury's award went to Williams's widow, Sandra, 30; their three children, ages 8, 6 and 5; his mother, and five sisters.

"They were an extremely close family," said Lovell. "I've seen a lot of people who have been injured, but this family was particularly devastated by this loss."

U.S.

Killer robot

Japanese worker first victim of technological revolution

TOKYO (UPI) The industrial robot that brought a technological revolution to Japan killed its first human last July when an auto worker was crushed to death by a machine he was fixing, labor officials said Tuesday.

"There's been a tendency to put aside the regulations in the labor standards law with these new machines," an official at the Labor Standards Bureau in the western province of Hyogo said.

There are some 50,000 industrial robots operating in Japanese plants, accounting for about 70 percent of the automatons in the world.

Kenji Urada, 37, an employee at the Akashi plant of Kawasaki Heavy Industries, Ltd. was pinned by a robot's arm against a machine for processing automobile gears and crushed to death.

The officials at the labor bureau said the accident took place July 4 as Urada was trying to check the malfunctioning machine, the officials said.

Urada apparently hit the onswitch accidentally after leaping over a chain fence built around the robot that was labeled "Off Limits," they said.

The fence was designed to automatically shut off the power supply when opened to repair the robot, they said.

Fellow workers who rushed to the scene could not help him as they did not know how to stop the machine, they said.

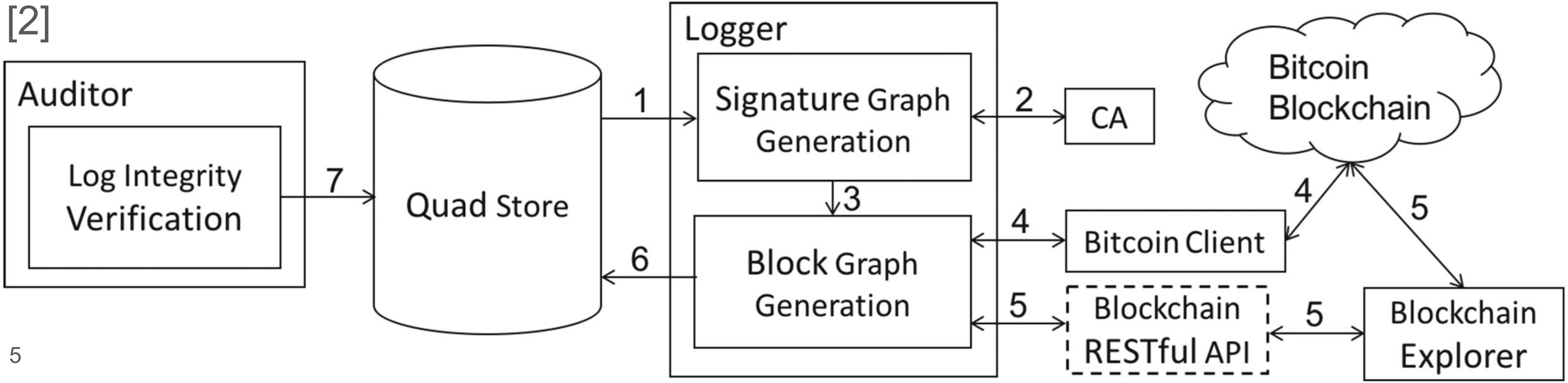
Related Work

[1] **Table 1.** Data immutabilized within the transaction output

Name	Bytes	Content	Description
$prefix$	2	"SL"	OP_RETURN prefix
$version$	1	1	Version of the data structure
$logId$	16	-	Unique log identifier
H_j	32	-	Hash of the checkpoint

- [1] Jordi Cucurull, and Jordi Puiggalí. "Distributed immutabilization of secure logs." International Workshop on Security and Trust Management. Springer, Cham, 2016.
- [2] Andrew Sutton, and Reza Samavi. "Blockchain enabled privacy audit logs." International Semantic Web Conference. Springer, Cham, 2017.

[2]



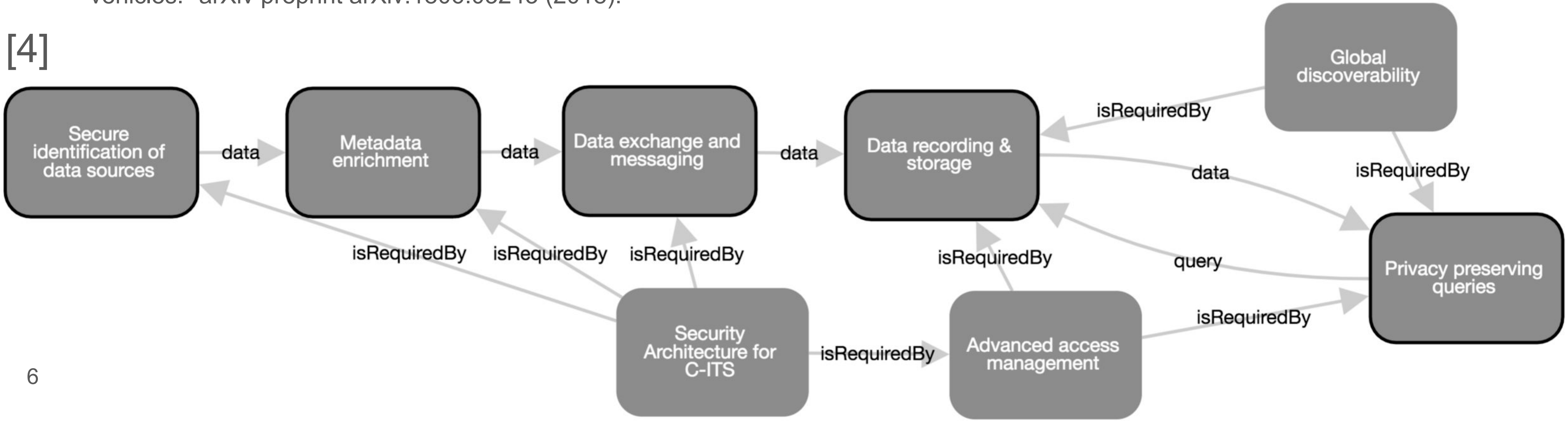
[3]

Related Work

- [3] Sebastian Taurer, Bernhard Dieber, and Peter Schartner. "Secure data recording and bio-inspired functional integrity for intelligent robots." 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018.

- [4] Viktoras Kabir Veitas, and Simon Delaere. "In-vehicle data recording, storage and access management in autonomous vehicles." arXiv preprint arXiv:1806.03243 (2018).

[4]



Constraints

CPSs such as commercially viable mobile robots incur contains in addition to just cyber security:

- Limited energy capacity
  - Efficiency is key to prolong deployment
  - Minimize computational overhead
- Build of Materials
  - Practical costs and availability at scale
  - Consumer grade hardware compatible
- Finite memory storage
  - Event data recording must be continuous
  - Maximize data retention for auditability
- Unreliable network connectivity
  - Uplink bandwidth a precious commodity
  - Flexible QoS to minimize overhead

Roles & Obligated Parties

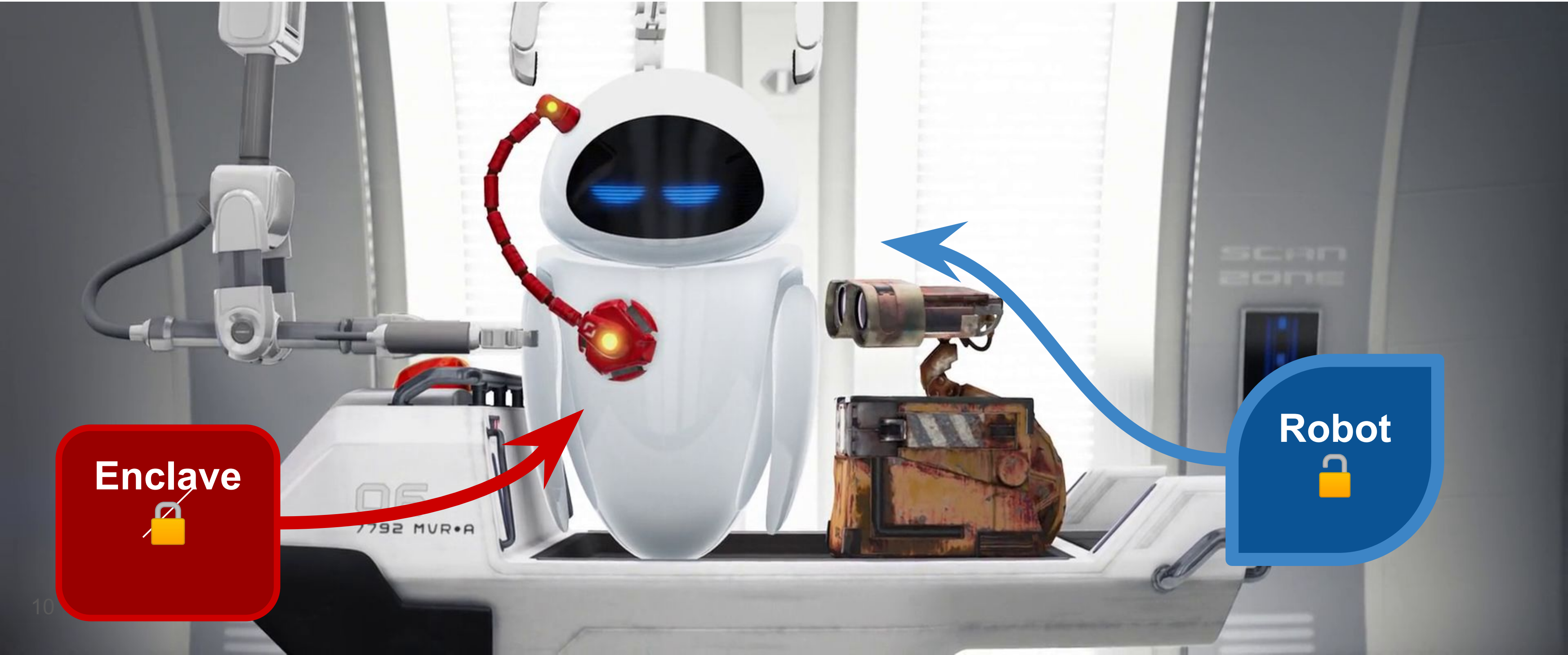
Custodian: Robot or autonomous vehicle OEM
Reporter: Trusted Logger or Recorder Enclave
Owner: End-User or Operator
Auditors: Regulatory Agencies or Gov.
Attacker: Malicious Actor

Party: Custodian

Obligated subject of log content and tasked with log preservation. e.g. Robot or autonomous vehicle OEM.

Party: Reporter

An independent party responsible for faithfully recording events. e.g. Trusted Logger or Recorder Enclave.



Party: Owner

Mediating party that has a stake in ensuring log integrity, authenticity, confidentiality. e.g. End-User or Operator.

Party: Auditor

Observing parties called upon to investigate and validate record archives. e.g. Regulatory Agencies or Gov.

Party: Attacker

Any party intent on modifying the record to falsify events.
e.g. Malicious actor expunging incriminating evidence.

Storage Requirements

1. Data provision conditions

   requires consent on behalf of the **Owner** who transitively controls the log assets tracked.

2. Fair and undistorted competition

   trust should be distributed and shared across all validators (a.k.a **Custodians**).

3. Data privacy and data protection

   the co-location of logs external to that of the **Custodian** must be prevented.

4. Tamper-proof access and liability

   integrity and authenticity of logs must derive from an independent **Reporter**.

5. Data availability economy

   health and transparency of logs are contingent upon giving **Auditors** appropriate access.

System Properties

1. Secure identification of physical data sources

   attestation between devices trusted by the **Custodian** and **Reporter**.
2. Metadata enrichment

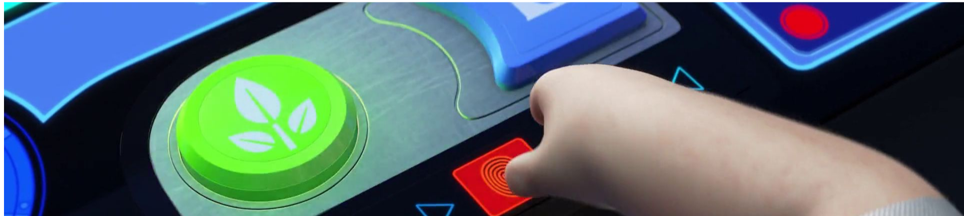
   log event context may be associated to respective **Owner**, **Custodian** and **Reporter** parties.
3. Data exchange and messaging

   authenticated encryption is used in establishing secure connectivity between parties.
4. Data recording & storage

   reporting remains flexible in terms of QoS as well as reasonable in resource consumption.
5. Access management

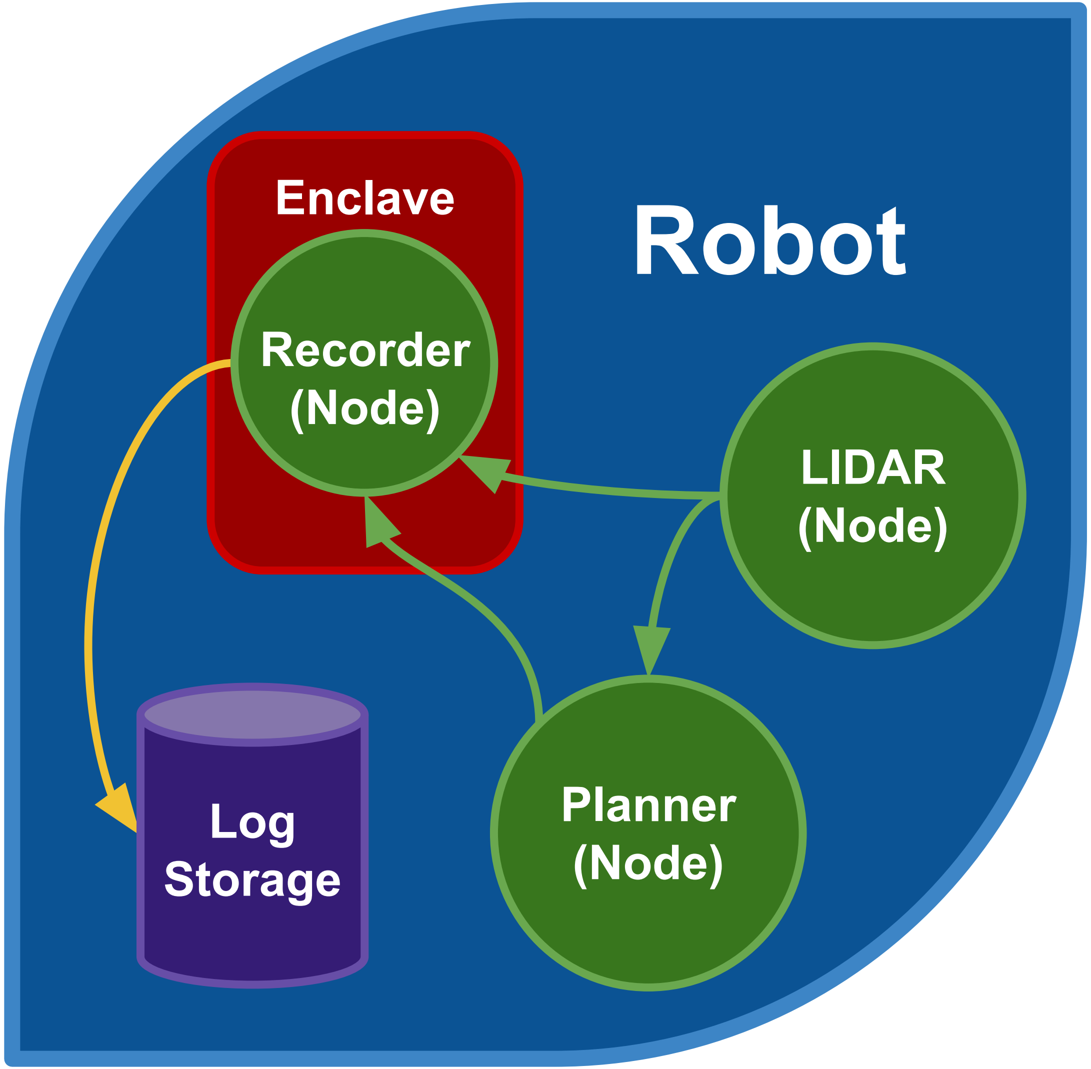
   rights, obligations, and authorization of parties must be explicitly defined and enforceable.



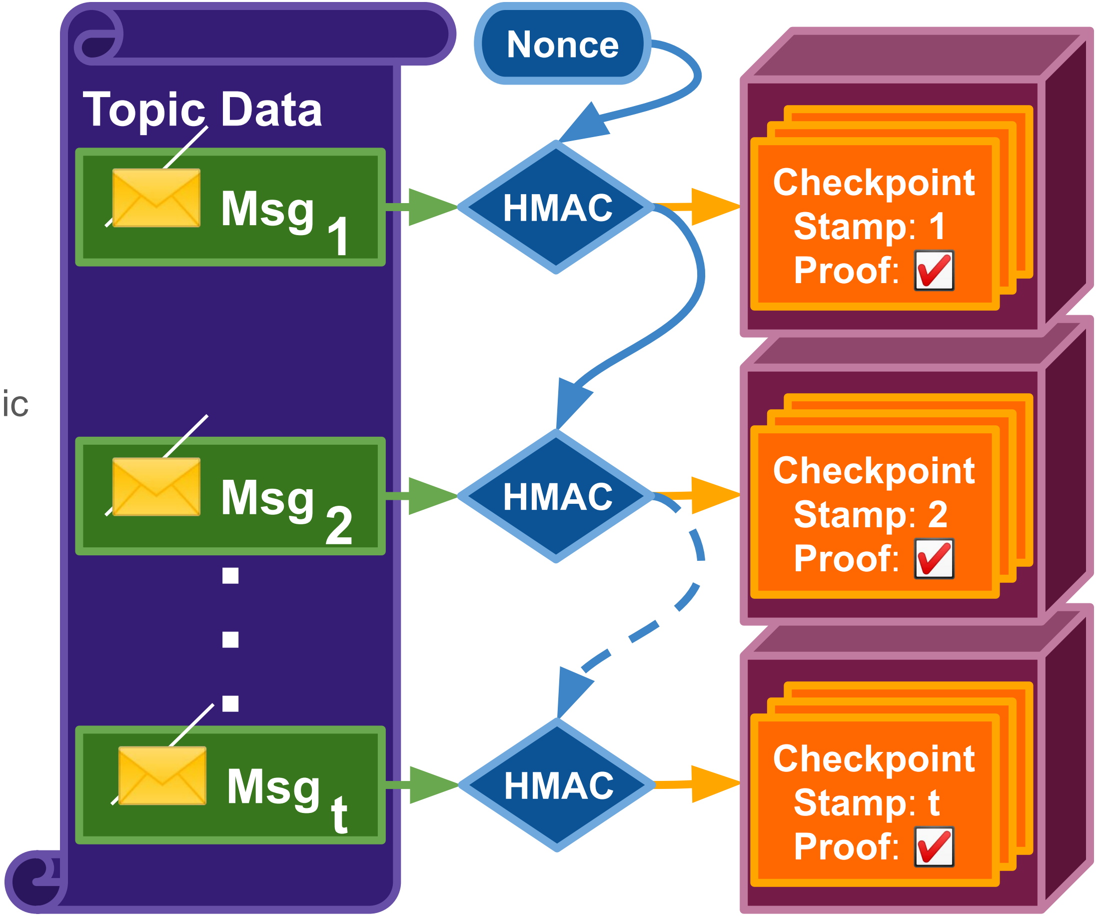
Approach

- Introducing Black Block Recorder (BBR)
- To mediate separation of concerns, only the recording process need be enclaved
- Enclaved process generates the logs by capturing broadcasted message traffic
- Before streaming logs to arbitrary storage, data must first be rendered immutable



Approach

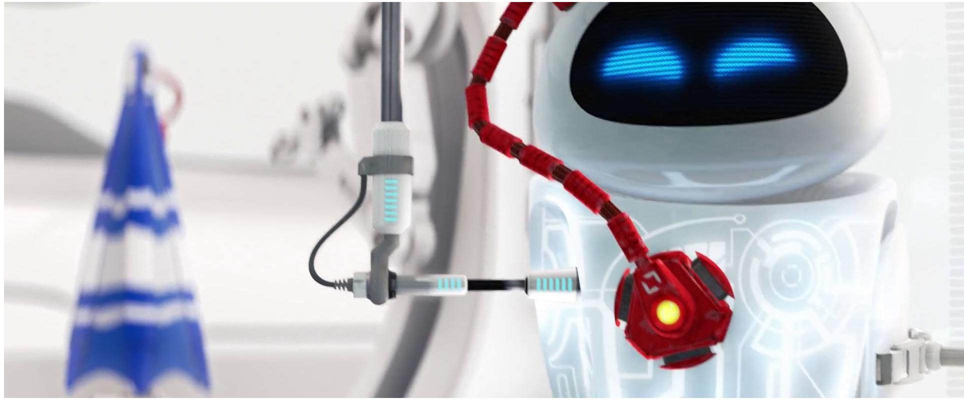
- Striding checkpoints are disseminated to the a blockchain for eventual log auditing
- Checkpoints are comprised of cryptographic linked integrity proofs that are indexed
- Thus we explore EDRs based upon Distributed Ledger Technology (DLT)

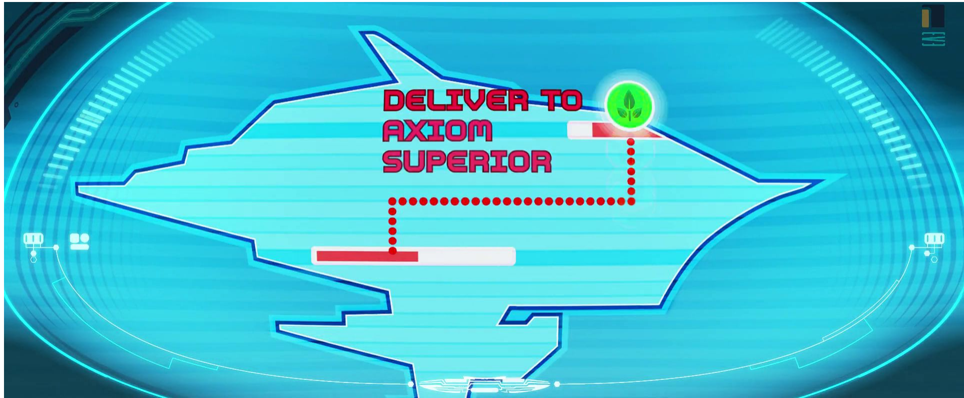


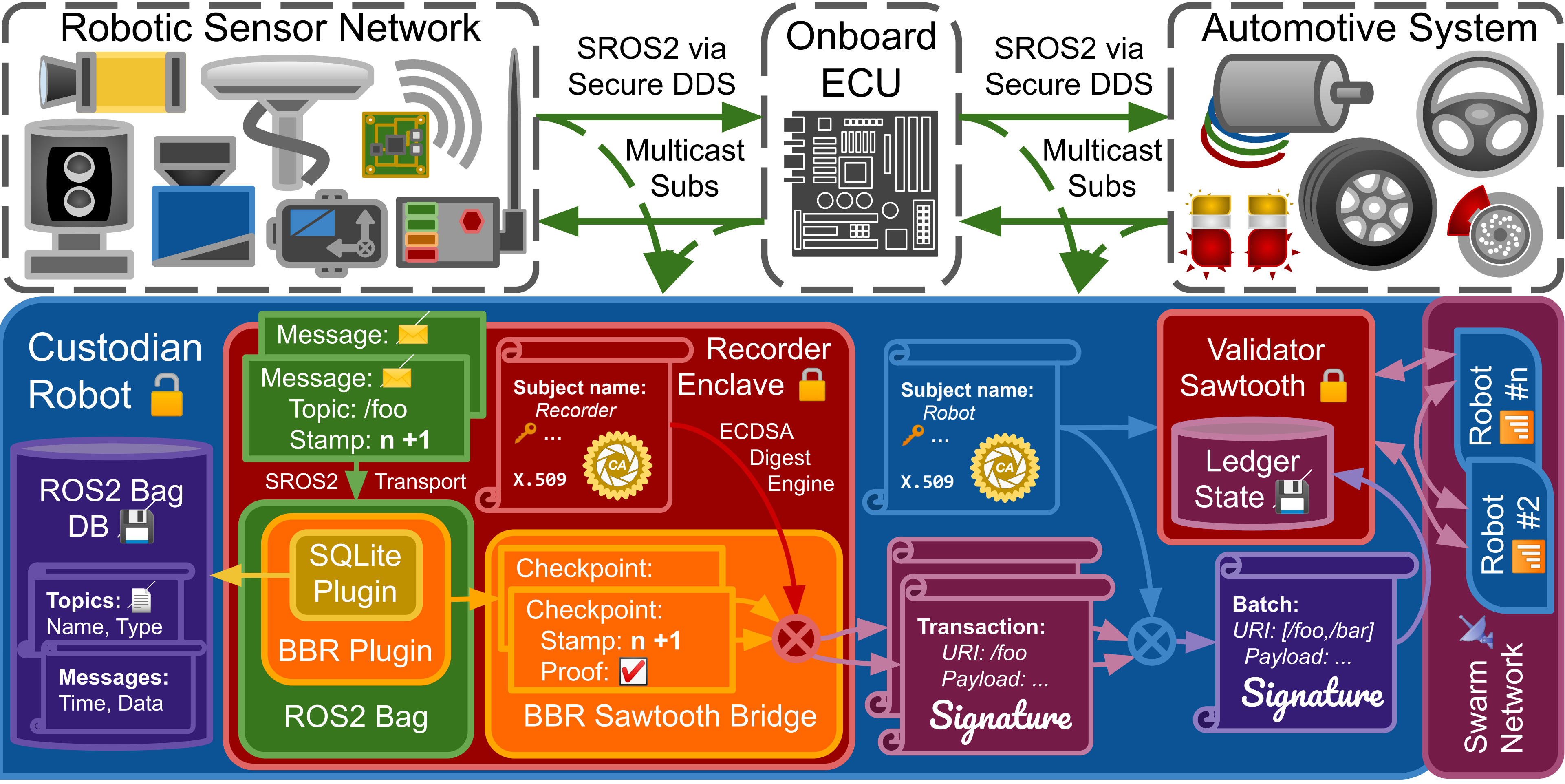
$$Chk_i = (i, h_i) \quad h_i = HMAC(h_{i-1}, LogMsg_i) \qquad (1)$$
$$where \quad h_0 \xleftarrow{\$} \{0,1\}^m$$

Information Flow

- While a robot platform is held suspect, an enclave (e.g. TTE) is reserved for the reporter
- Event data is received within the enclave via SROS2 using authenticated encryption
- Linked integrity proofs for each asset tracked are batched and signed via BBR rosbag plugin
- Batches are then signed within the enclave, then co-signed by the robot's via BBR bridge
- Thus an append-only forgery to DLT record would necessitate the collusion of both the custodian and its respective assigned reporter



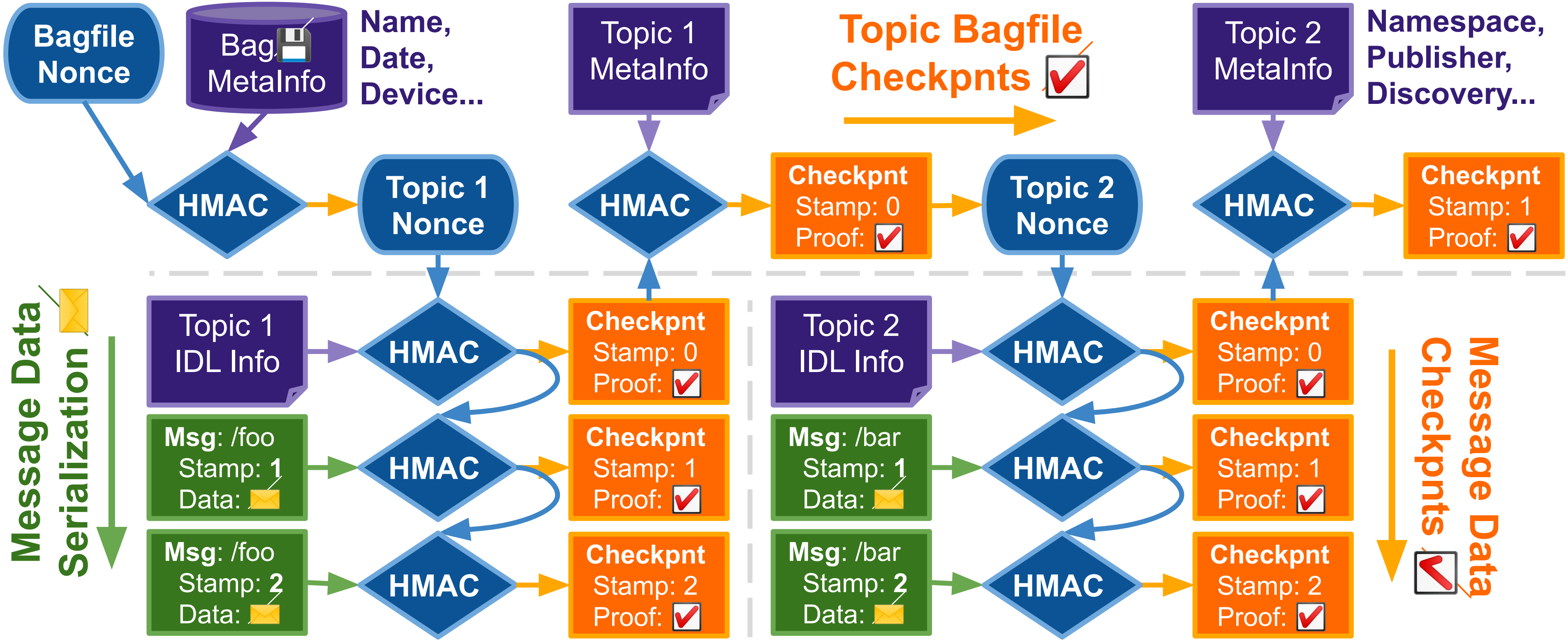




Event Data ➔ Enclave ➔ ROS2 Bag Plugin ➔ Bridge ➔ DLT Validator ➔ DLT Network

Schema Structure

- Recorded event data is structured using ROS2bag files and compatible SQLite schema
- Data insertions into the database are achieved via 2D-array hash-chains of checkpoints
- Where the primary axis checkpoints each bagfile topic's genesis-block and meta-info
- While the secondary axis checkpoints the insertion of respective message data
- This coupling affords a holistic integrity proof of the entire database, while preserving topics as a time series atomic, fostering parallelism



Bagfile Nonce
Bag MetaInfo
Name, Date, Device...
Topic 1 MetaInfo
Topic Bagfile Checkpnts
Topic 2 MetaInfo
Namespace, Publisher, Discovery...
HMAC
Topic 1 Nonce
HMAC
Checkpnt Stamp: 0 Proof:
Topic 2 Nonce
HMAC
Checkpnt Stamp: 1 Proof:
Message Data Serialization
Topic 1 IDL Info
HMAC
Checkpnt Stamp: 0 Proof:
Msg: /foo Stamp: 1 Data:
HMAC
Checkpnt Stamp: 1 Proof:
Msg: /foo Stamp: 2 Data:
HMAC
Checkpnt Stamp: 2 Proof:
Topic 2 IDL Info
HMAC
Checkpnt Stamp: 0 Proof:
Msg: /bar Stamp: 1 Data:
HMAC
Checkpnt Stamp: 1 Proof:
Msg: /bar Stamp: 2 Data:
HMAC
Checkpnt Stamp: 2 Proof:
Message Data Checkpnts

Topics Schema

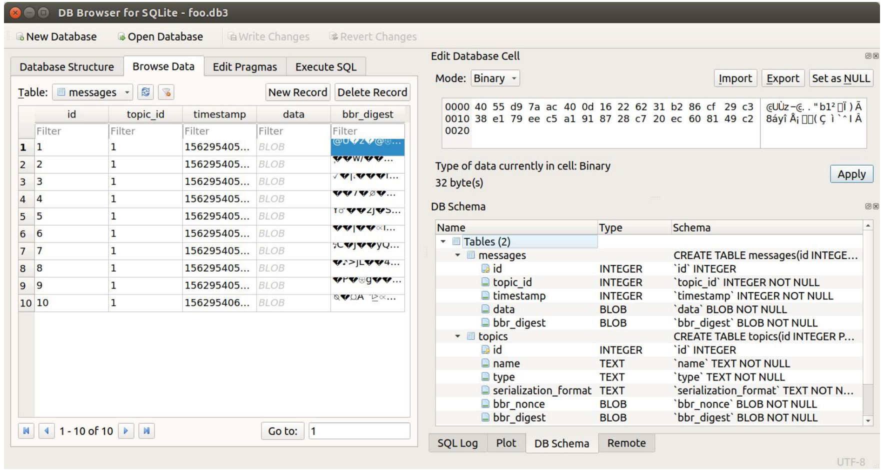
- The nonce and digest for each topic are retained to verify even IDL info & metadata
- Here we'll choose to hash the topic name metadata separate from the IDL info
- This allows for message data to remain verifiable even when want to remap topics
- As a superset of the default SQLite schema, bags from the BBR plugin remain compliant

Topics				BBR	
id	name	type	format	nonce	digest
1	/foo	gps	rtps	*bits*$_A$	*bits*$_{A0}$
2	/bar	imu	rtps	*bits*$_B$	*bits*$_{B0}$
3	/baz	lidar	rtps	*bits*$_C$	*bits*$_{C0}$

$$bits_{bag} \xleftarrow{\$} \{0,1\}^m \quad (2)$$
$$bits_A \leftarrow HMAC(bits_{bag}, Proto(name_{bag})) \quad (3)$$
$$bits_{A_0} \leftarrow HMAC(bits_A, Proto(type_A, format_A)) \quad (4)$$
$$bits_B \leftarrow HMAC(bits_{A_0}, Proto(name_A)) \quad (5)$$
$$bits_{B_0} \leftarrow HMAC(bits_B, Proto(type_B, format_B)) \quad (6)$$
$$bits_C \leftarrow HMAC(bits_{B_0}, Proto(name_B)) \quad (7)$$
$$bits_{C_0} \leftarrow HMAC(bits_C, Proto(type_C, format_C)) \quad (8)$$

Messages Schema

- Each message is checkpointed by the time of arrival and data payload received
- Here we'll choose not to hash the topic ID, given message is bound to the topic digest



Messages				BBR
id	topic	time	data	digest
1	id 1	*int64*	*bits*	*bits*$_{A1}$
2	id 2	*int64*	*bits*	*bits*$_{B1}$
3	id 1	*int64*	*bits*	*bits*$_{A2}$

$$bits_{A_1} \leftarrow HMAC(bits_{A_0}, Proto(time_{A_1}, data_{A_1})) \quad (9)$$

$$bits_{B_1} \leftarrow HMAC(bits_{B_0}, Proto(time_{B_1}, data_{B_1})) \quad (10)$$

$$bits_{A_2} \leftarrow HMAC(bits_{A_1}, Proto(time_{A_2}, data_{A_2})) \quad (11)$$

- This allows for message data to remain verifiable even when want to splice bags
- Checkpoints link to previous message of the same topic, not necessarily previous insert

Smart Contract

- Checkpoints ensure immutability, but not authenticity and non-repudiation by itself
- Smart Contracts (SC) encapsulate access control logic for state in permissioned DLTs
- Digital Asset Modeling Language (DAML) is used to formalize contract specification
- Record/Checkpoint data structures define the base abstractions used in SC modeling

```
data Record = Record with  -- data type struct
    r_name: Text    -- "/sensors/exteroceptive/gps"
    r_type: Text    -- "gps"
    r_format: Text -- "rtps"
    r_nonce: Text   -- "<bits_A>"
    r_digest: Text -- "<bits_A_0>"
    r_checkpoints: [Checkpoint] -- monotonic list
```

```
data Checkpoint = Checkpoint with -- data type struct
    c_proof: Text -- "<bits_A_1>"
    c_stamp: Int   -- 1
```

Smart Contract

- An EDR SC is formed from parties involved voluntarily entering as obligated signatories
- External parties may also provisioned observational access to the SC state
- Control for creating new records, to retain checkpoints, is delegated to the reporter

```
template Edr -- Smart Contract for EDRs
 with
   auditors: [Party] -- Regulatory Agencies
   custodian: Party  -- Robot/Vendor Identity
   owner: Party      -- User/Operator Identity
   reporter: Party   -- Logger/TEE Identity
 where
   signatory custodian, owner, reporter -- obligated
   observer auditors -- non-obligated parties
   ensure unique (custodian :: owner :: reporter)
   controller reporter can -- create many Records
     non consuming Edr_Record : ContractEd Edr Record
       with record: Record
       do create Record with edr = this; record
```

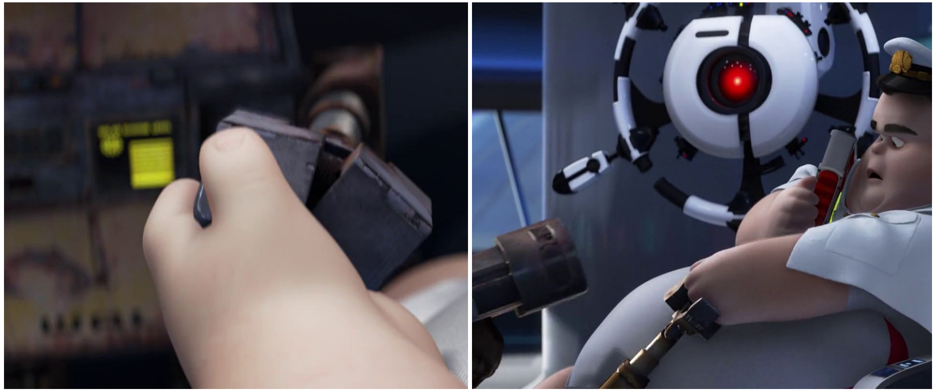
Smart Contract

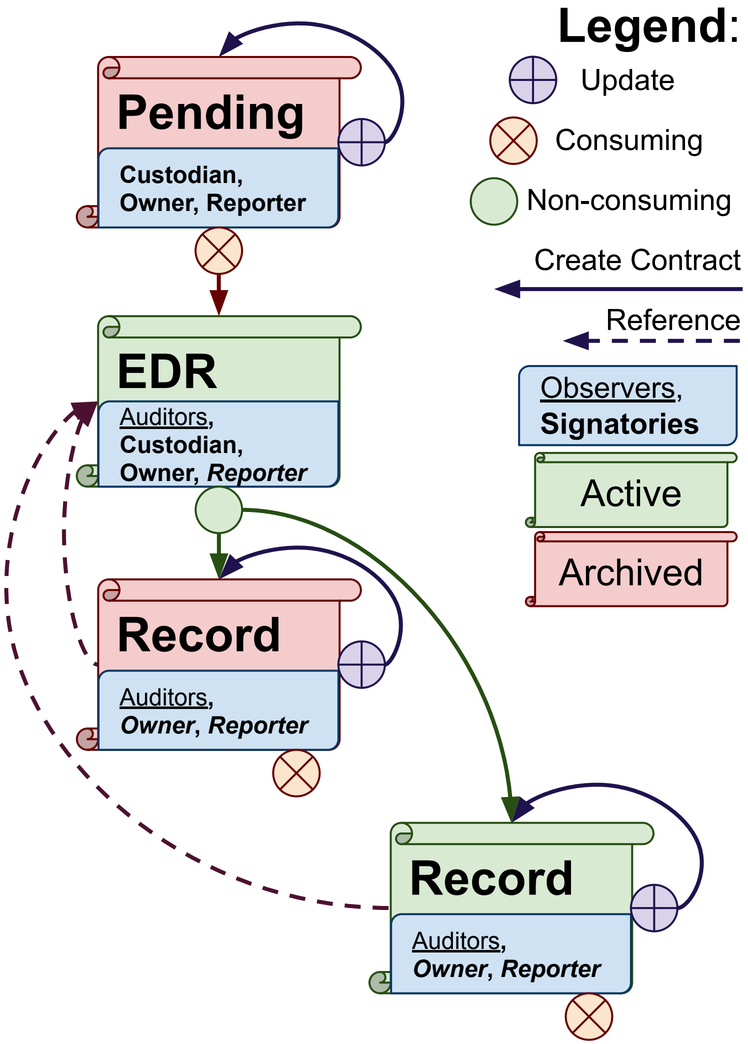
- An Record SC is formed from a reference to the EDR SC and initial record struct object
- Only the **Reporter** is permitted the choice to append checkpoints into the ledure state
- Valid checkpoints are appended by asserting stamps remain monotonically increasing
- Both **Reporter** and **Owner** are permitted the choice to permanently finalize the record

```
template Edr Record -- Smart Contract for Records in EDR
 with
   edr: Edr       -- Reference EDR of origin
   record: Record -- Initial Record state
 where
   signatory edr.owner, edr.reporter
   observer edr.auditors -- custodian can be excluded
   choice EdrRecord_Append : ContractEd Edr Record
       with checkpoints: [Checkpoint] -- [] for batching
       controller edr.reporter -- Only Reporter appends
       do let -- Update Record with added checkpoints
           is_valid = check Monotonic record checkpoints
           _record = append Checkpoints record checkpoints
         assert (is_valid == True) -- Error on invalid
         create Record with edr; record = _record
   choice EdrRecord_Finalize : () -- Archives Contract
     controller edr.owner, edr.reporter
       do return () -- Finalized Record is un-appendable
```

Smart Contract

- An EDR SC is preliminarily proposed via pending contract used to collect the necessary multi-party signatories.
- The pending SC is consumed to create the agreed EDR SC
- The **Reporter** may create multiple referencing records and append checkpoints, that itself or the **Owner** may finalize





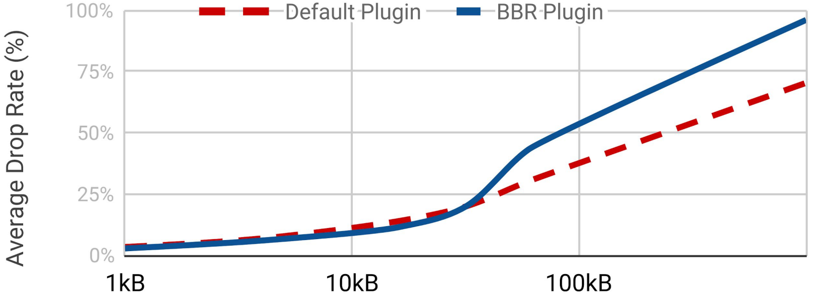
Results

- Drop Rate & Load performance comparison of BBR storage/bridge plugins with regard to ROSBag2's default SQLite plugin.
- Benchmarked via ROS2 Crystal, 2.6GHz Intel i7-6700HQ, with RTI Connext RMW on loopback interface.

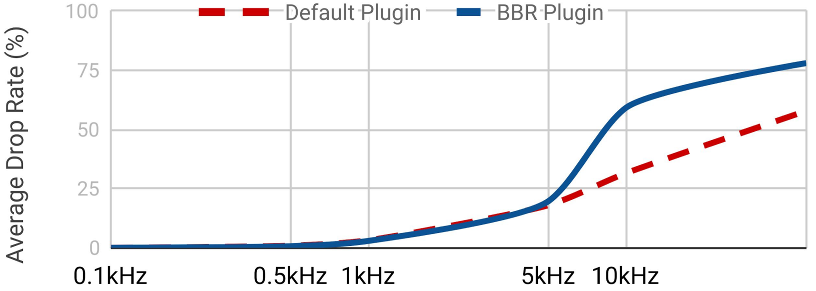
Payload	Size (bytes)	Requirements
Chk_i	36	Ledger Disk Storage
Signed Transaction	≥629	Network Bandwidth
Signed Batch	≥965	Network Bandwidth

TABLE I: BBR Payload Allocations

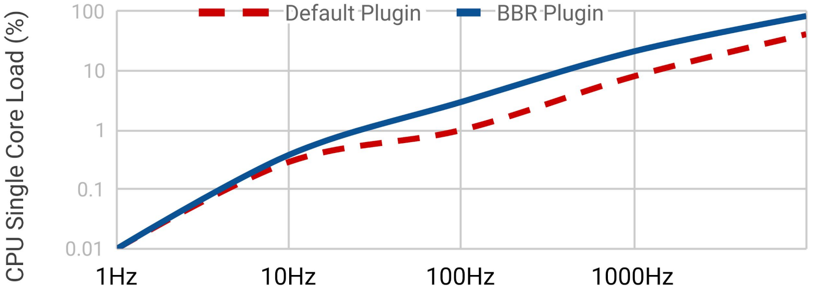
Msg. Size vs Drop Rate @ fixed 1kHz Msg. Freq.



Msg. Freq. vs Drop Rate @ fixed 1kB Msg. Size



Msg. Freq. vs CPU Load @ fixed 1kB Msg. Size



Conclusion

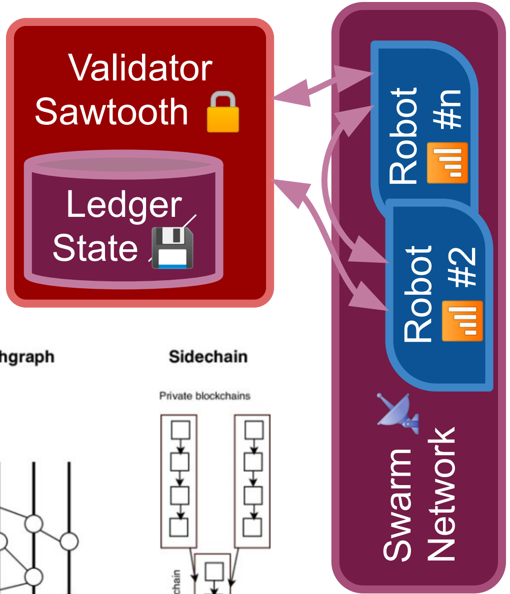
- Engineering trade offs always exist
  - Security vs. Performance
  - Immutability vs. Drop-Rate
- However trade offs can be balanced
  - Adjusting QoS for sensitive topics
  - Sparsity via Striding checkpoint
- Promising uses of DLTs in Cryptobotics
  - EDR applications leveraging distributed consensus
  - When Byzantine Fault Tolerance is advantageous

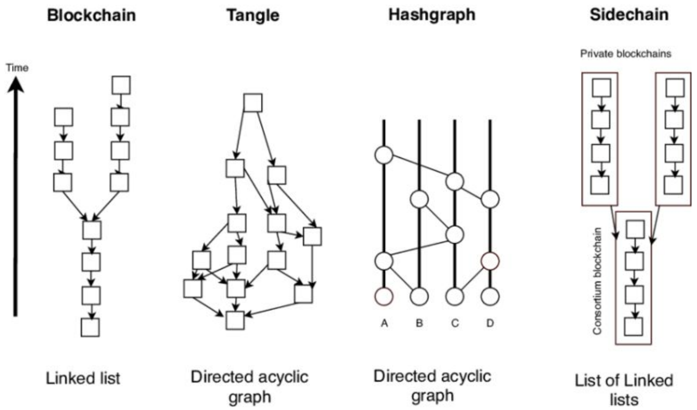
Messages			BBR
id	topic	...	digest
1	id 1	...	$bits_{A1}$
2	id 1	...	$bits_{A2}$
3	id 1	...	$bits_{A3}$
2	id 1	...	$bits_{A4}$

Future Work

- Benchmark with consensus
  - Profile validator overhead
  - Characterize scale limitations
- Explore alternative DLT pardimes, e.g:
  - Directed Acyclic Graphs vs Blockchains
  - E.g. IOTA's Tangle
- Record at the middleware level (DDS)
  - Capture all traffic like services/actions
  - E.g. Parameter updates, planner feedback

IOTA

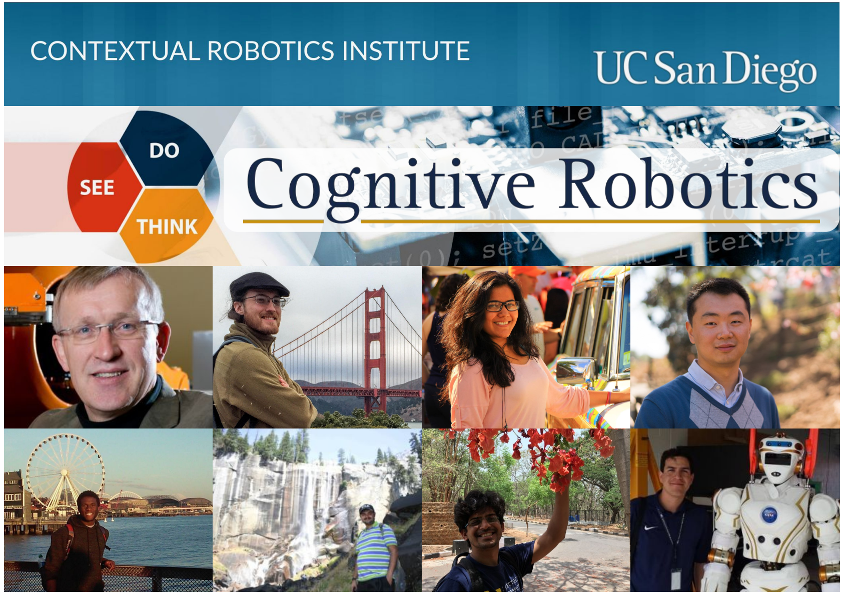




Acknowledgements

THE LINUX FOUNDATION

HYPERLEDGER
BLOCKCHAIN TECHNOLOGIES FOR BUSINESS



Università Ca'Foscari Venezia

Advances in Autonomous,
Distributed and Pervasive systems

Links

Current Slides: roscon.ros.org/2019/presentations/ROSCon2019_Black_Block_Recorder.pdf

GitHub

dledr/bbr_ros2

Black Block Recorder with ROS2 | Immutable Logging via Blockchain - dledr/bbr_ros2

GitHub

ros2/sros2

tools to generate and distribute keys for SROS 2. Contribute to ros2/sros2 development by creating an account on GitHub.

Hyperledger

Hyperledger Sawtooth – Hyperledger

A modular platform for building, deploying, and running distributed ledgers.

Hyperledger

Hyperledger – Open Source Blockchain Technologies

Hyperledger is a multi-project open source collaborative effort hosted by The Linux Foundation, created to advance cross-industry blockchain technologies.

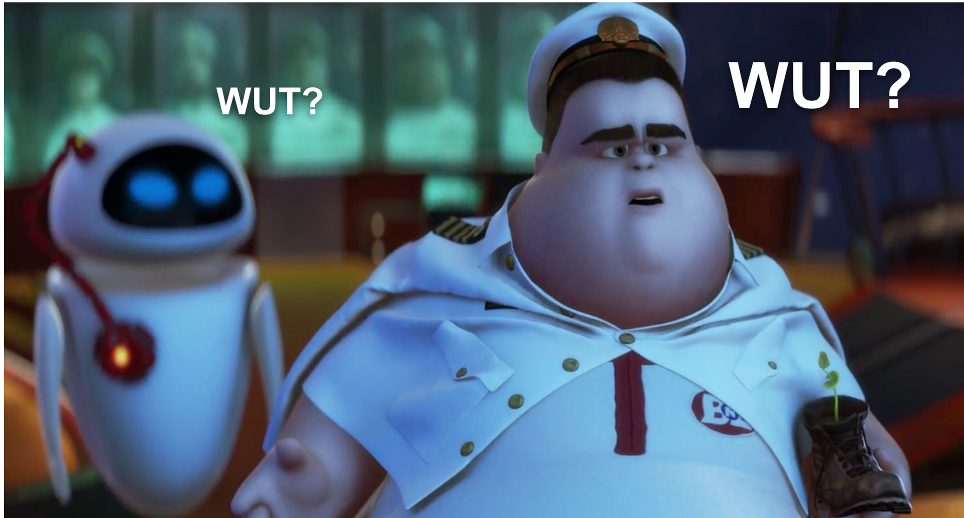
GitHub

digital-asset/daml

The DAML smart contract language. Contribute to digital-asset/daml development by creating an account on GitHub.

Questions?

A113: Full Autopilot Override
BBR: **B**lack **B**lock **R**ecorder
BFT: **B**yzantine **F**ault **T**olerance
CPS: **C**yber-**P**hysical **S**ystem
DAG: **D**irected **A**cyclic **G**raph
DFI: **D**igital **F**orensic **I**nvestigation
DLT: **D**istributed **L**edger **T**echnology
DSA: **D**igital **S**ignature **A**lgorithm
EDR: **E**vent **D**ata **R**ecorder
MAC: **M**essage **A**uthentication **C**ode
ROS: **R**obot **O**perating **S**ystem
SROS: **S**ecure **ROS**



References

R. White, G. Caiazza, A. Cortesi, Y. I. Cho and H. I. Christensen, "Black Block Recorder: Immutable Black Box Logging for Robots via Blockchain," in *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3812-3819, Oct. 2019. doi: [10.1109/LRA.2019.2928780](https://doi.org/10.1109/LRA.2019.2928780)

“WALL-E”, Pixar Animation Studios, Walt Disney Studios Motion Pictures, (film) Jun. 2008.

M. Luthi, A. Sparrow, et al. “WALL-E”, in BOOM! Studios, (comic) Nov. 2009 - Jun. 2010.

More about:

Ruffin: about.me/ruffin

Gianluca: about.me/caiazza