CORTX & FDMI

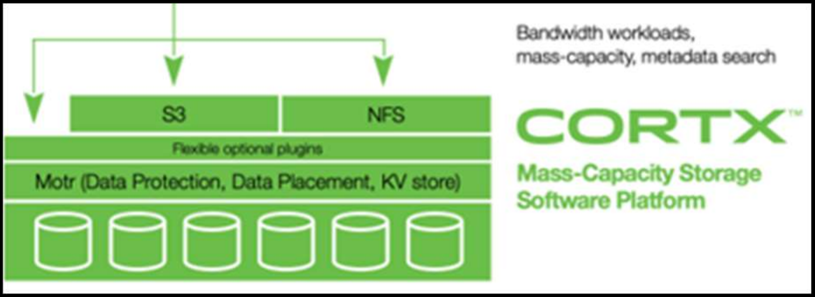
CORTX Meet An Architect (4th November 2021)

Liana Valdes Rodriguez
Ph.D. Student / Seagate Intern
Florida International University

Outline

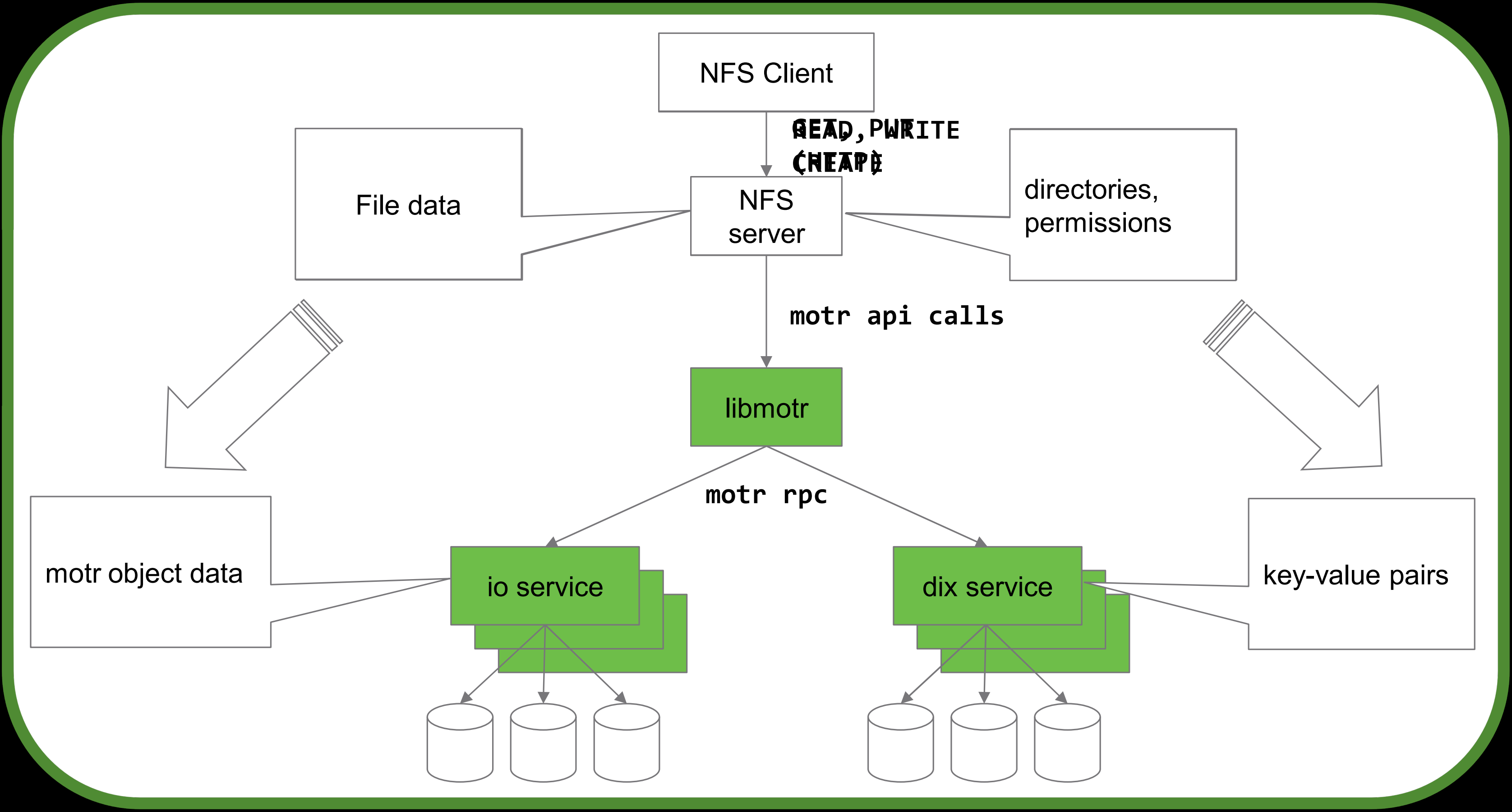
CORTX Introduction

- CORTX is a distributed object storage system designed for:
  - Great efficiency
  - Massive capacity
  - Storage device flexibility (NVMe, SSD, HDD)
  - High HDD-utilization
- Complete 100 % Open Source
- Highly Resilient:
  - High tolerance for hardware failure
  - Faster recovery times using Network Erasure Coding
- **Extremely Horizontal Scalable**


Bandwidth workloads, mass-capacity, metadata search
S3
NFS
Flexible optional plugins
Motr (Data Protection, Data Placement, KV store)
CORTX™
Mass-Capacity Storage Software Platform

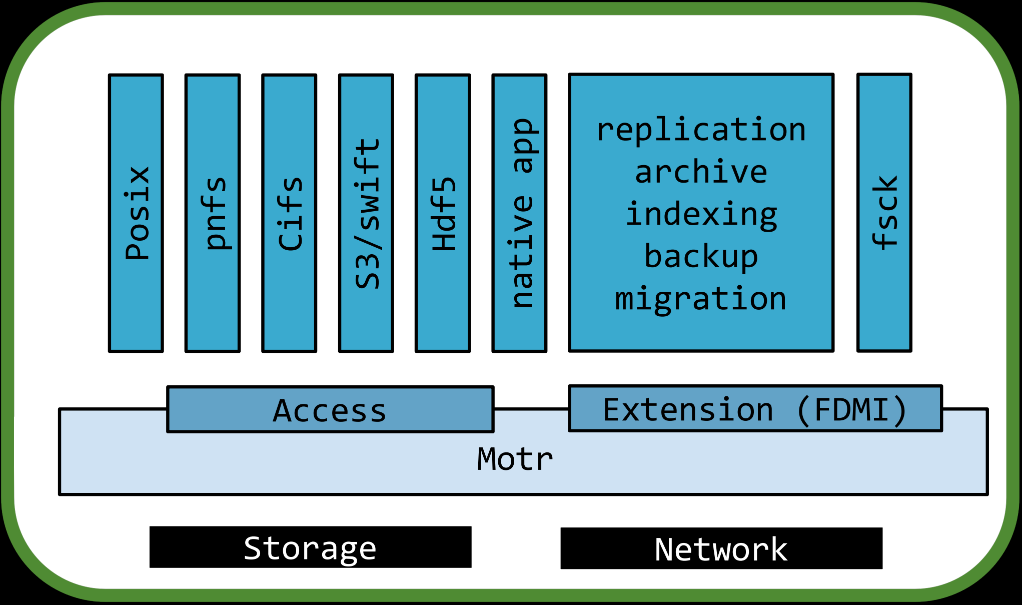
Source/Footnote:

CORTX Architecture: Data Flow



What is Motr?

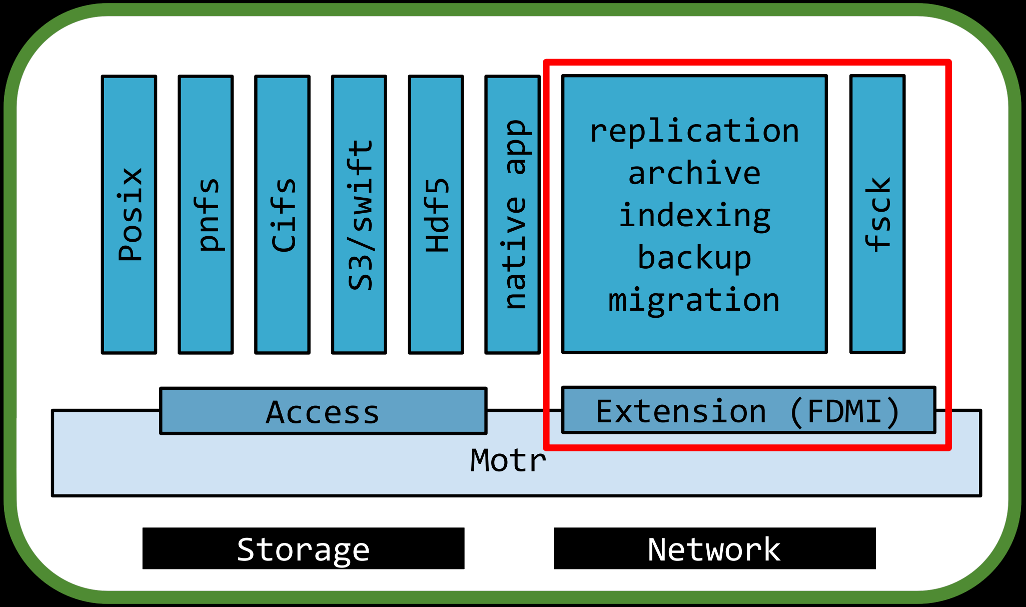
- ❑ Motr is a core component of CORTX for the new Cloud Software Stack developed by Seagate.
- ❑ Through its client interface Motr provides:
  - **Object store**
  - **Key-value store**
- ❑ The software stack includes other components:
  - S3 / pNFS server
  - Administration and monitoring toolset
- ❑ Similar products:
  - Ceph
  - ActiveScale
  - IBM/COS



Source/Footnote:

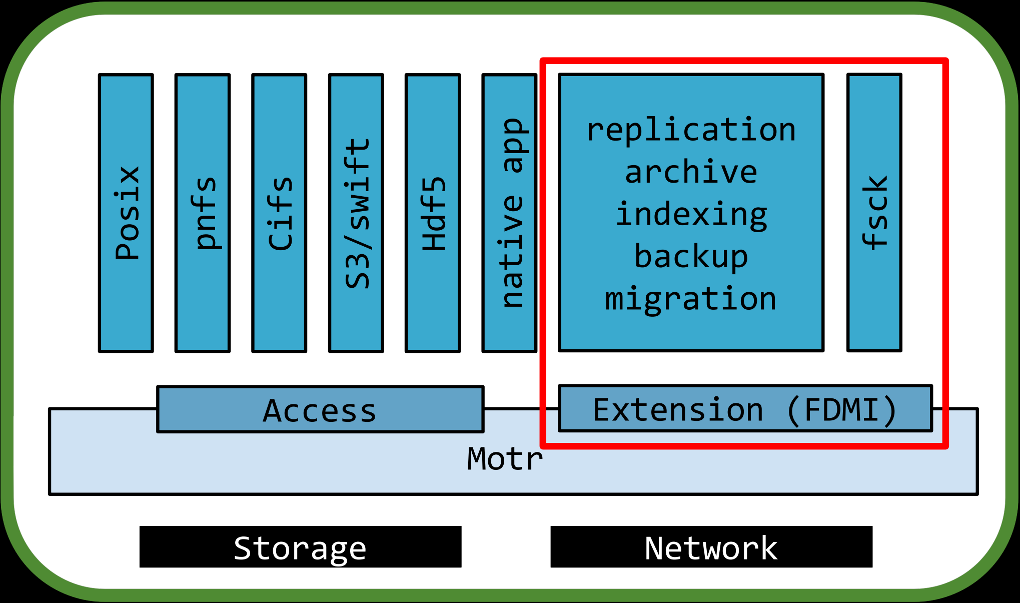
Why FDMI is needed?

- ❑ Need to include new features and capabilities without having to natively modify the core base.
- ❑ Add flexibility to developers for easily modifying and deploying in CORTX.
- ❑ Scalable since allows to add functionalities on multiple servers.
- ❑ Lessons learnt from previous distributed storage systems design such as Lustre.



FDMI Overview

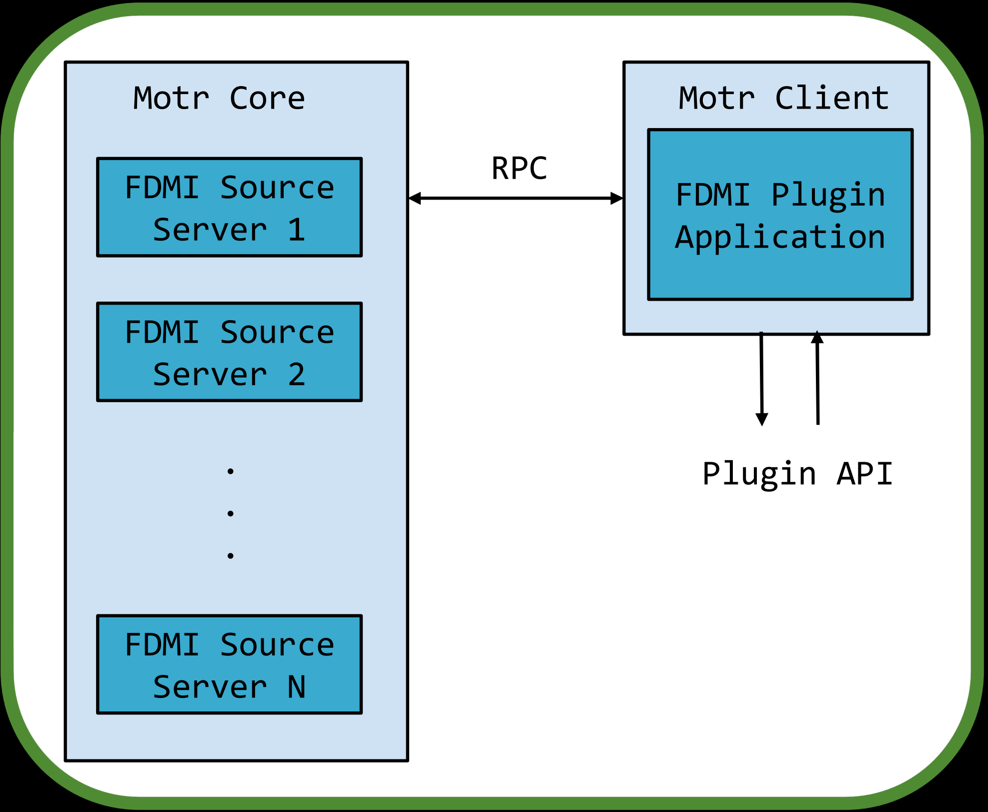
- FDMI: File Data Manipulation Interface (Motr Extension interface)
- Extensions: migration, backup, archive, replication, compression, encryption, indexing, tiering, defragmentation, scrubbing, format conversion, re-striping, *etc*.
- Extensions are:
  - Developed independently (without modifications to the core code), possibly by 3rd parties
  - Deployed independently (on additional nodes, without compromising fast path)
  - Scalable
  - Reliable (transactionally coupled with the core)



Source/Footnote:

How FDMI works?

- ❑ FDMI is a scalable publish-subscribe interface.
- ❑ Each Motr instance produces cross-referenced records describing operations.
- ❑ FDMI plugin registers a filter in the Motr Filter Database, that selects some records.
- ❑ Each instance sends matching records to the plugin (in batches) with their transactional contexts.
- ❑ A plugin/app acts on records and sends acknowledgements back to the source instances.



FDMI example plugins

01 Integrity Checking

02 Asynchronous Replication

03 Function Shipping

04 Ransomware Detection

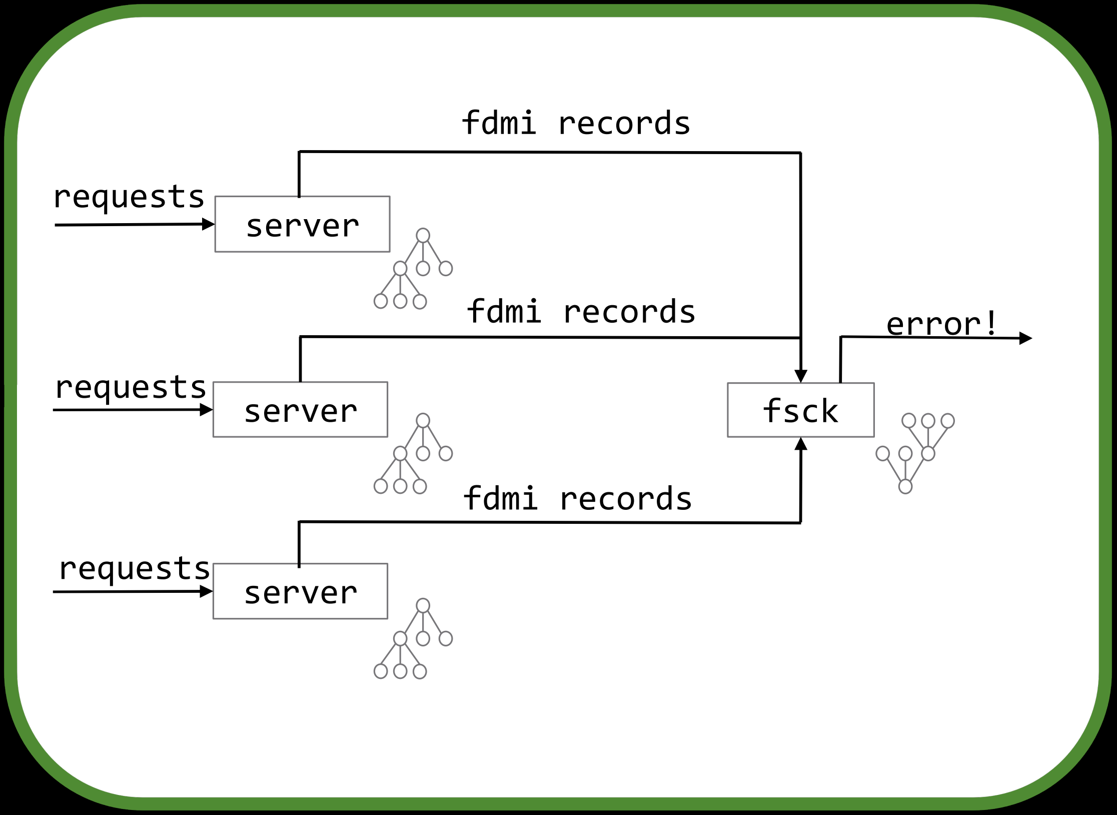
05 Hackathon Word Count

FDMI example plugin: Integrity Checking

- ❑ How to recover from catastrophic failures?
- ❑ Traditional fsck tool
  - Not distributed
  - Specific to the meta-data format
  - Does not scale:
    - Time
    - Space
- ❑ Need scalable integrity checking
  - Distributed fsck solution: Motr Recovery mechanism
  - Run it all the time, on dedicated separate nodes (horizontal scalability)
  - Update meta-data to match system evolution (through FDMI publish-subscribe)
  - Detect inconsistencies, report and recover from redundancy

Source/Footnote:

FDMI example plugin: Integrity Checking



Traditional meta-data persistent structures:

- ❑ Block allocation
- ❑ Key distribution
- ❑ B-tree structure

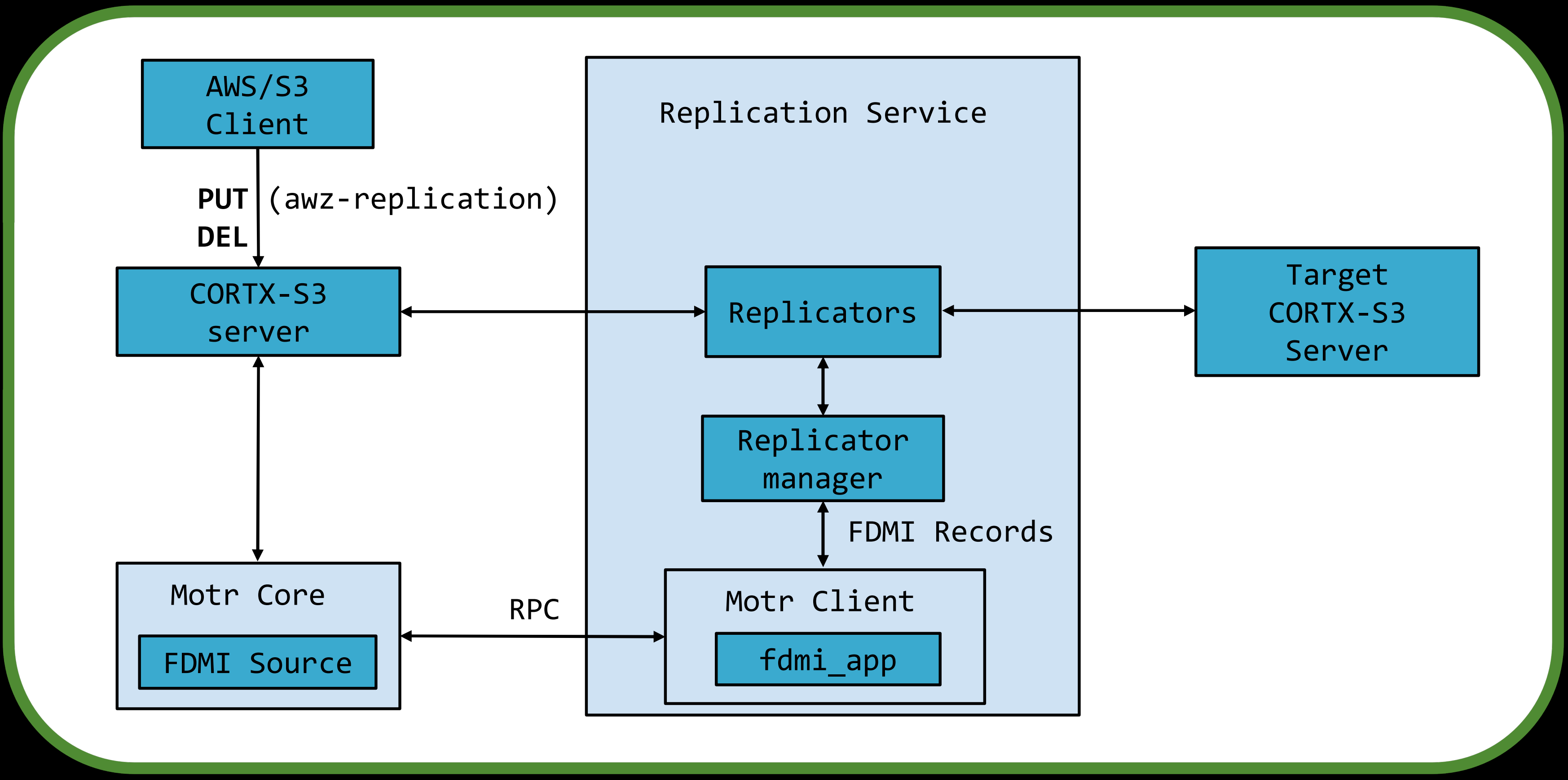
FDMI Example Plugin: Replication

Asynchronous Multisite Replication

- ❑ S3 server processes async replication requests for objects using per bucket level replication policy and set specific replication metadata field per object that needs to be replicated.
- ❑ Motr FDMI application listens for metadata updates with the specific replication metadata field and generates events.
- ❑ Replication manager picks up the replication request in the form of FDMI records and communicates with Replicators.
- ❑ Replicators make a copy of the object to the target S3 endpoint using S3 APIs.

Source/Footnote:

FDMI Example Plugin: Replication



Source/Footnote:

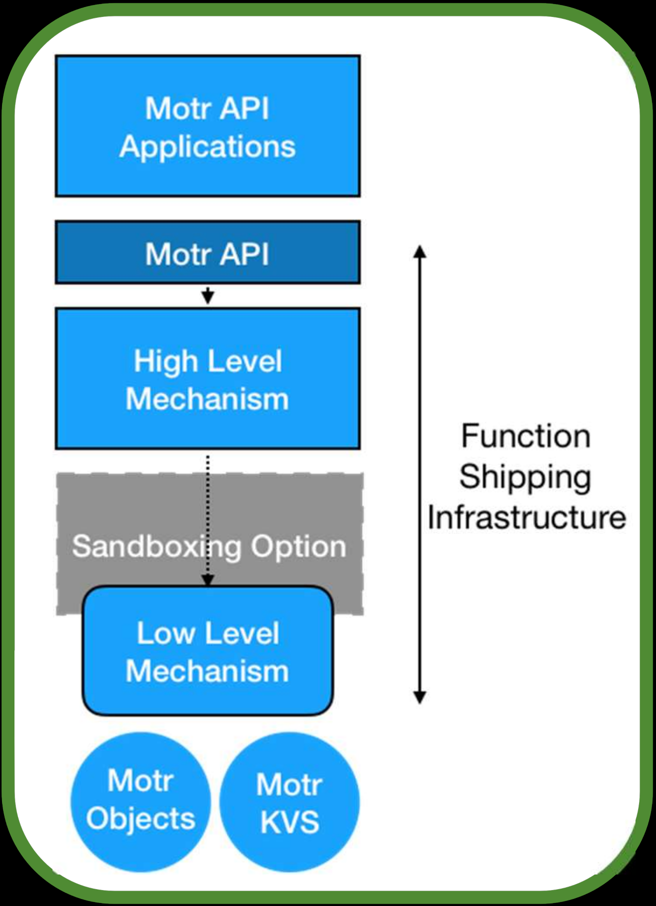
FDMI Example Plugin: Function Shipping

Mechanism can be used by **external applications** as well as **internally within Motr** (Ex: Calculations for distributed parity)

The different mechanisms of the implementation are:

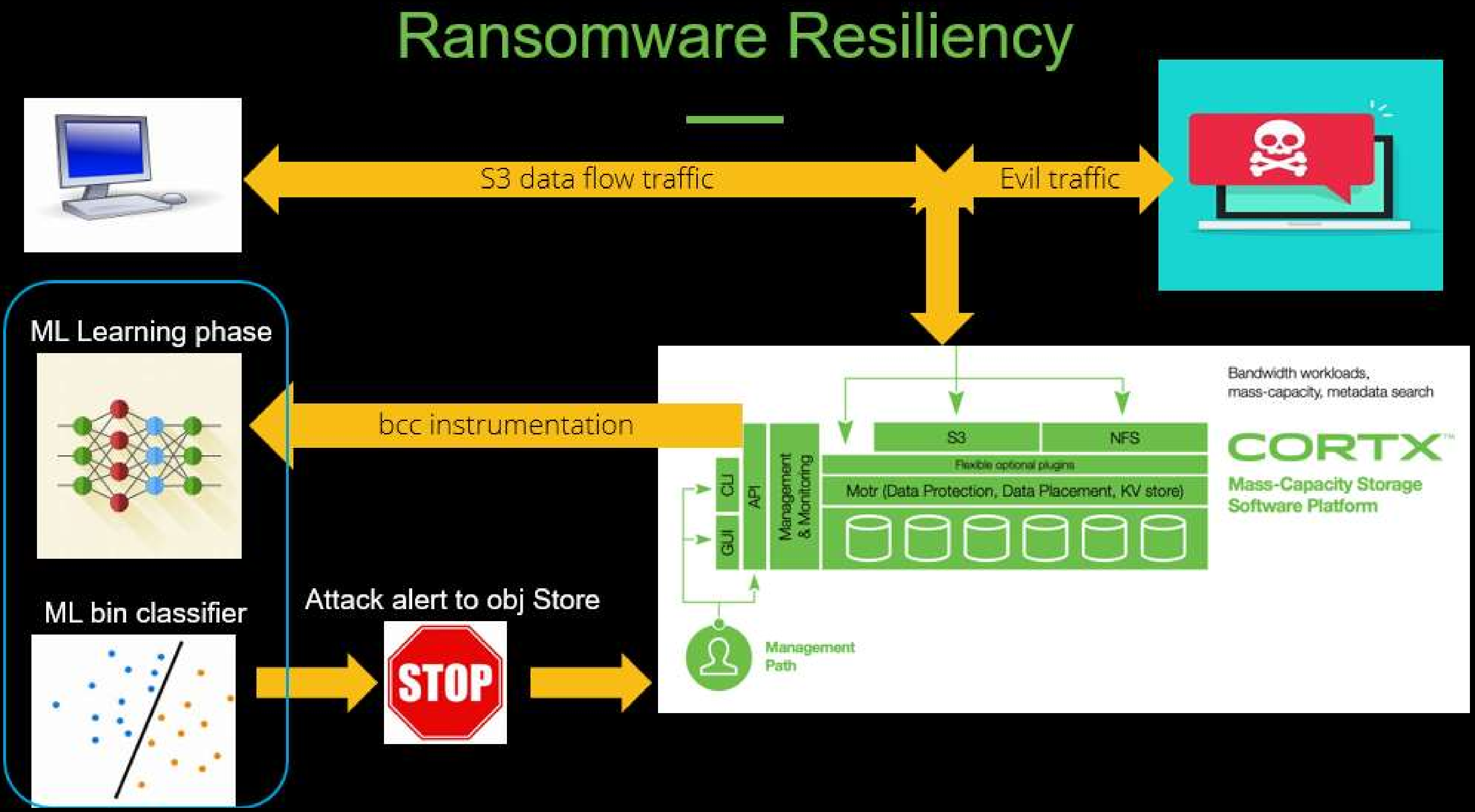
1. Low-level mechanism: This is provided to “register” and invoke a trusted function on a particular node.
2. Sand-boxing: This provides the ability to run untrusted functions in separate protected domains.
3. High-level mechanism: This mechanism does the actual invocation of functions against objects and indices (use layouts) using the low-level and optionally the sandboxing mechanism.

- Potential use of FDMI to subscribe to Function Output Streams as FDMI sources.
- Dedicated FDMI application handling in-storage computations



Source/Footnote:

FDMI example plugin: Ransomware



Ransomware Resiliency
S3 data flow traffic
Evil traffic
ML Learning phase
bcc instrumentation
ML bin classifier
Attack alert to obj Store
STOP
Bandwidth workloads, mass-capacity, metadata search
CORTX
Mass-Capacity Storage Software Platform
S3
NFS
Flexible optional plugins
Motr (Data Protection, Data Placement, KV store)
CLI
GUI
API
Management & Monitoring
Management Path

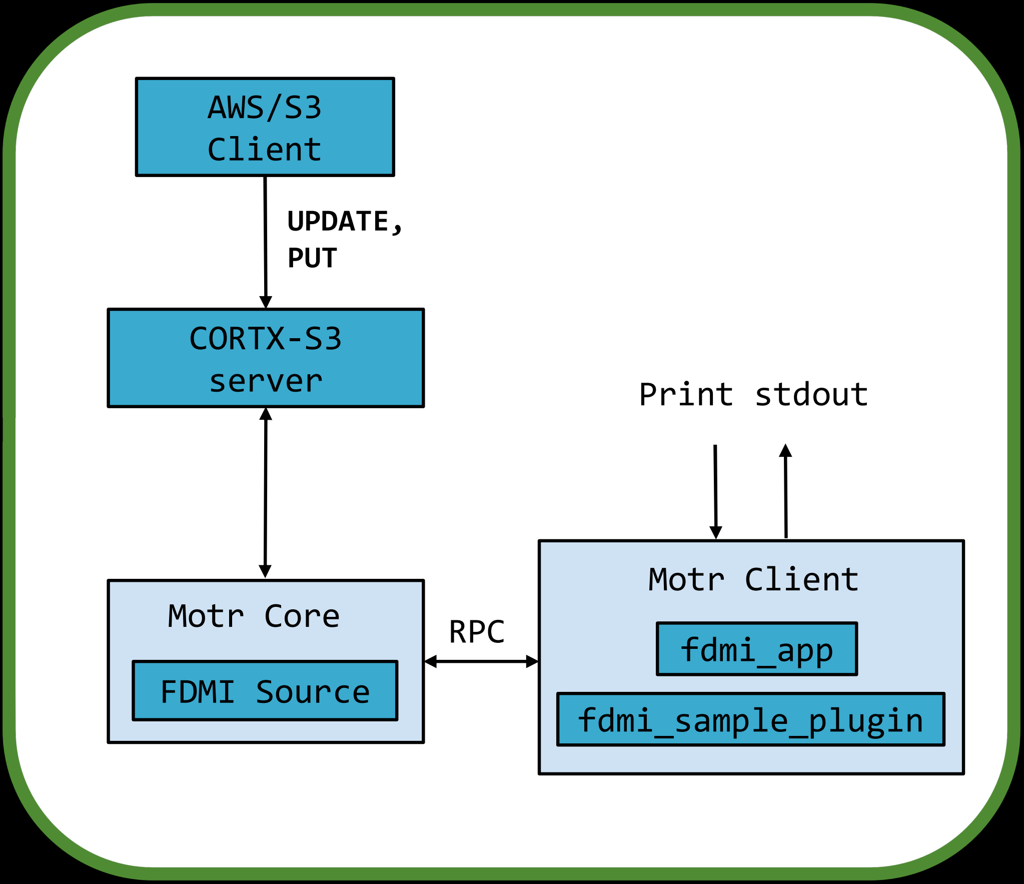
Source/Footnote:

FDMI Example Plugin: Hackathon

Object Data Word Count Application

- ❑ FDMI sample plugin executable using Motr client API that prints to stdout records of new S3 update requests.
- ❑ Filters FDMI substrings (Object-name, Bucket-name) are specified at the CORTX Motr initialization time when setting the cluster.
- ❑ FDMI python wrapper that reads from stdout and process the records.
- ❑ For each new update request, we need to connect to AWS client and get the specific object data.
- ❑ Count the number of words in the object data and print the word and count of the 40 most popular words to stdout.

FDMI Example Plugin: Hackathon



Source/Footnote:

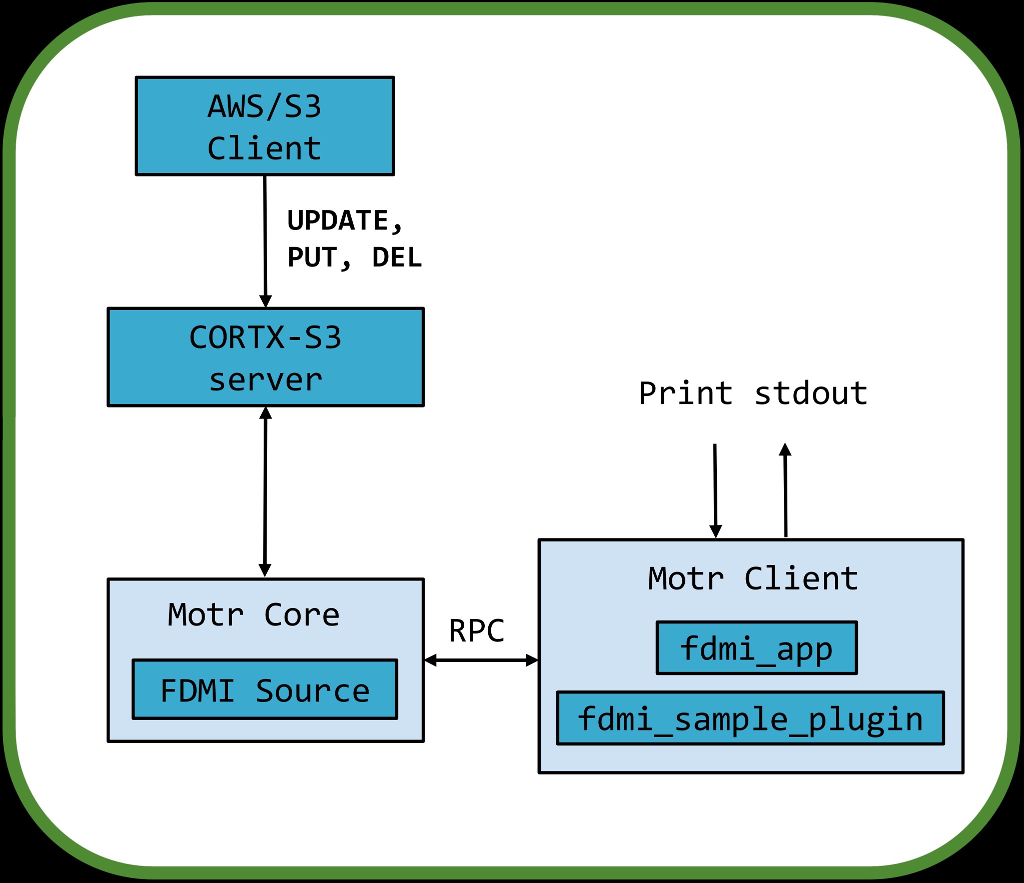
FDMI Example Plugin: Hackathon

- ❑ Link to Github Page with more details for the Hackathon event:

https://cortx.link/UCD

- ❑ Link to demo video:

https://lia54.github.io/cortx-fdmi-app/images/Liana%20Valdes%20FDMI%20app%20demo%20video-20211014_105759-Meeting%20Recording.mp4



Source/Footnote:

Questions?

Thank You!

I would like to thanks John Bent, Sai Narasimhamurthy, Ganesan Umanesan, Nikita Danilov, Paul Heath and Rupasree Roy for sharing and improving the content of this presentation.
Special Thanks to the ADG team for helping me through out all the work and keeping up with my questions.