Hardware Accelerated Blockchain Operations

April 29, 2021

Today’s Presenters

Olga Buchonia

Chair of SNIA Blockchain TWG. Extensive experience in executive leadership, engineering and management, research and development, leadership and mentoring, test and problem-solving.

olga@myactionspot.com

Parmeshwr Prasad

Co-Chair of SNIA Blockchain committee. Passionate about newer technologies.

Parmeshwr.Prasad@dell.com

Agenda

- Blockchain Understanding
- Blockchain in Storage
- SNIA Blockchain Storage Technical Work Group Work (TWG)
- SmartNIC Use Case in Blockchain
- SCM Helping Blockchain Features

SNIA Legal Notice

- The material contained in this presentation is copyrighted by SNIA unless otherwise noted.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced in their entirety without modification
  - The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of SNIA.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion, please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

  NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

About SNIA

- The Storage Networking Industry Association is a not-for-profit global organization, made up of member companies spanning the global storage market. SNIA's mission is to lead the storage industry worldwide in developing and promoting standards, technologies, and educational services to empower organizations in the management of information. To this end, SNIA is uniquely committed to delivering standards, education, and services that will propel open storage networking solutions into the broader market.

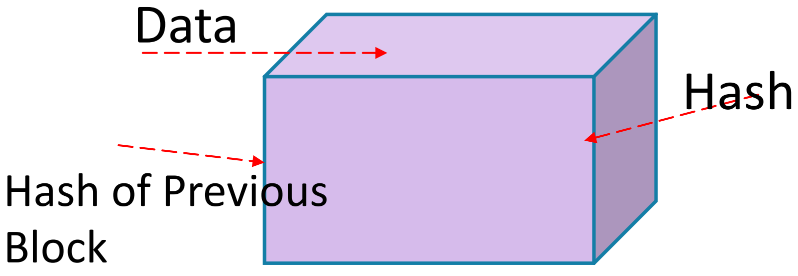
SNIA-at-a-Glance

180 industry leading organizations

2,500 active contributing members

50,000 IT end users & storage pros worldwide

Blockchain Understanding



- Blockchain is a distributed database of records stored in blocks.
- Blockchain is secured using peer validation in cryptography.
- Blockchain as a technology has several facets that directly or indirectly can impact user depending on implementation.

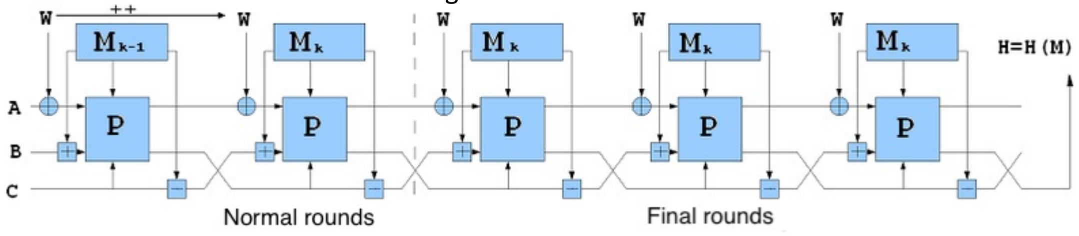
Blockchain, Hash and Consensus protocol

- Blockchain can use different cryptographic hash algorithms such as SHA-256 (one of the most popular), Whirpool, RIPEMD (RACE Integrity Primitives Evaluation Message Digest), Dagger-Hashimoto and others).
- Mercle tree is a blockchain construct which allows to build a chain by using hashes and data blocks.
- Consensus protocols is protocol for decision making such as Proof of Work, Proof of Space, Proof of Stake and etc. Each consensus protocol is using the distributed ledger to make a record for the block of data transferred.

Blockchain in Storage Applications Today

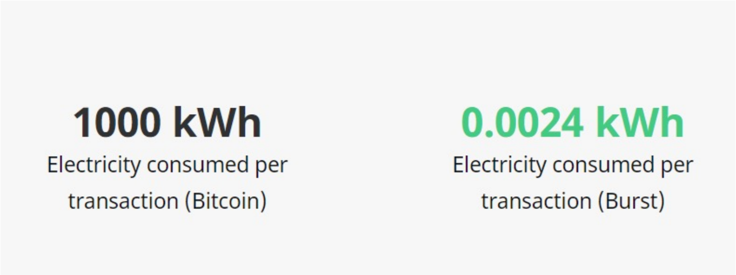
- Proof of Capacity uses the outputs of the shabal-256 cryptographic function to validate capacity to be used in mining.
- Shabal-256 currently is ASIC-resistant due to the IO requirements (as it requires writes).
- One time hashing process(plotting) versus continuous hashing.
- Mining process only involves reading the plots every new block(~ 4 min. average) and submitting the answers plus deadline(time to read to actual nonce).
- Power requirements for reading the plots greatly reduce overall energy consumed by the burstcoin blockchain.

Shabal Algorithm



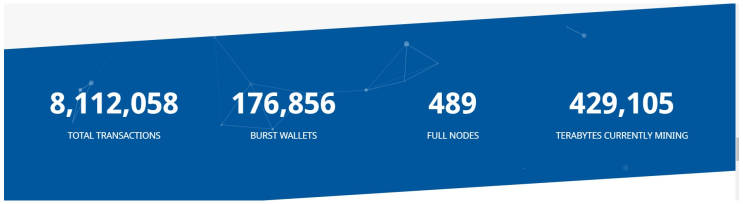
PROOF OF SPACE

- **Proof of space** (**PoSpace**), also called **Proof-of-capacity** (**PoC**), is a means of showing that one has a legitimate interest in a service (such as sending an email) by allocating a non-trivial amount of memory or disk space to solve a challenge presented by the service provider.
- Proof of space are very similar to proof of work, except that instead of computation, storage is used. Proof-of-space is related to, but also considerably different from, memory-hard functions and proofs of retrievability.
- After the release of Bitcoin, alternatives to its PoW mining mechanism were researched and PoSpace was studied in the context of cryptocurrencies.
- Proofs of space are seen as a fairer and greener alternative due to the general-purpose nature of storage and the lower energy cost required by storage.



1000 kWh
Electricity consumed per transaction (Bitcoin)

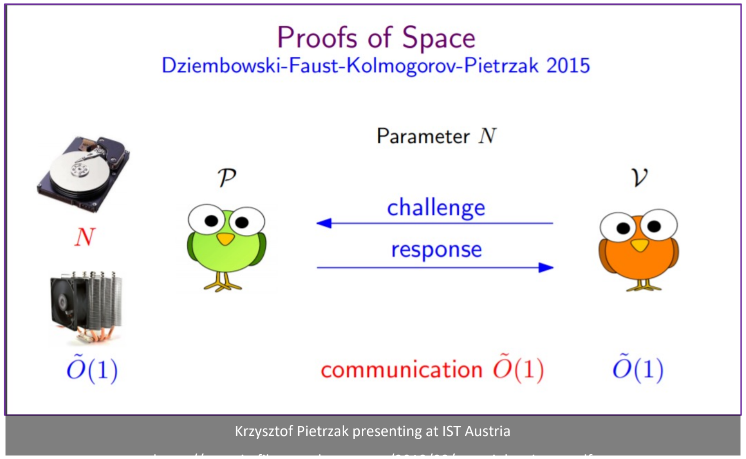
0.0024 kWh
Electricity consumed per transaction (Burst)



8,112,058	176,856	489	429,105
TOTAL TRANSACTIONS	BURST WALLETS	FULL NODES	TERABYTES CURRENTLY MINING

HOW IT WORKS – PROOF OF SPACE ?

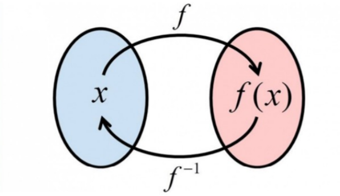
- A proof-of-space is a piece of data that a prover sends to a verifier to prove that the prover has reserved a certain amount of space.
- For practicality, the verification process needs to be efficient, namely, consume a small amount of space and time.
- For soundness, it should be hard for the prover to pass the verification if it does not actually reserve the claimed amount of space.
- Way to implement:
  - One way of implementing PoSpace is by using hard-to-pebble graphs.
  - The verifier asks the prover to build a labeling of a hard-to-pebble graph.
  - The prover commits to the labeling.
  - The verifier then asks the prover to open several random locations in the commitment.



Proofs of Space

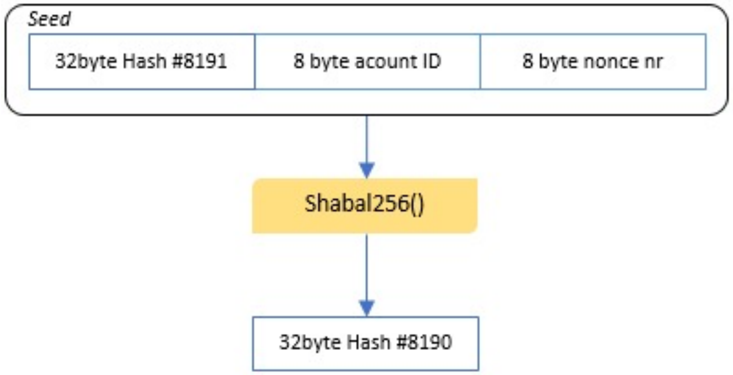
Dziembowski-Faust-Kolmogorov-Pietrzak 2015

Krzysztof Pietrzak presenting at IST Austria



BLOCKCHAIN AND PROOF OF CAPACITY

- Plotting is the process of generating plot files, which are just files storing a large number of pre-computed hashes. Each *plot* file contains one of more groups of 8192 hashes, these groups are called *nonces.* A nonce is exactly 256KB in size (8192 x 32 bytes per hash). Additionally, each nonce is divided into 4096 pairs of hashes, the pairs are referred to as *scoops*. Each nonce can also be identified by its index number, ranging from 0 to 2^64.

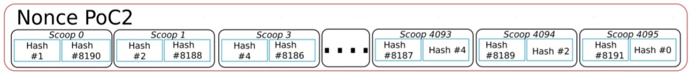


Seed

32byte Hash #8191 | 8 byte acount ID | 8 byte nonce nr

Shabal256()

32byte Hash #8190



Nonce PoC2

Scoop 0: Hash #1, Hash #8190

Scoop 1: Hash #2, Hash #8188

Scoop 3: Hash #4, Hash #8186

. . . .

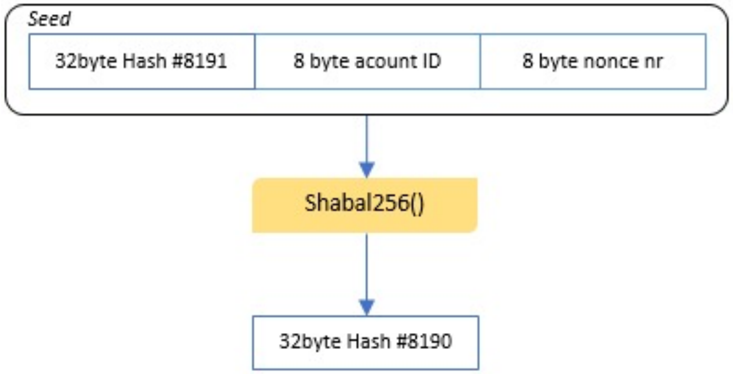
Scoop 4093: Hash #8187, Hash #4

Scoop 4094: Hash #8189, Hash #2

Scoop 4095: Hash #8191, Hash #0

BLOCKCHAIN AND PROOF OF CAPACITY

- Plotting is the process of generating plot files, which are just files storing a large number of pre-computed hashes. Each *plot* file contains one of more groups of 8192 hashes, these groups are called *nonces.* A nonce is exactly 256KB in size (8192 x 32 bytes per hash). Additionally, each nonce is divided into 4096 pairs of hashes, the pairs are referred to as *scoops*. Each nonce can also be identified by its index number, ranging from 0 to 2^64.



Seed

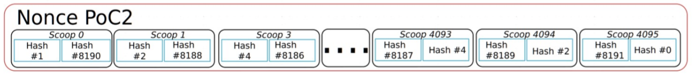
32byte Hash #8191

8 byte acount ID

8 byte nonce nr

Shabal256()

32byte Hash #8190



Nonce PoC2

Scoop 0: Hash #1, Hash #8190

Scoop 1: Hash #2, Hash #8188

Scoop 3: Hash #4, Hash #8186

. . . .

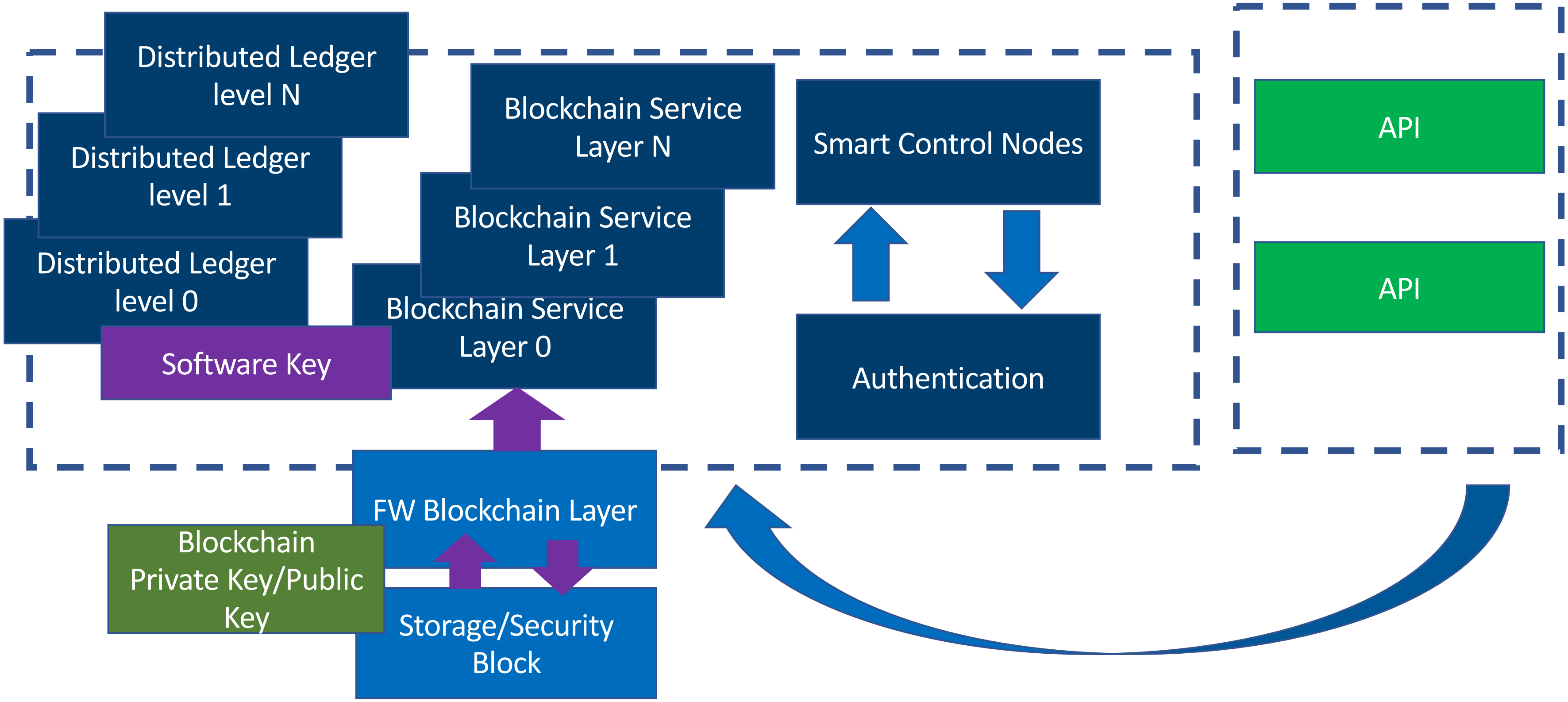
Scoop 4093: Hash #8187, Hash #4

Scoop 4094: Hash #8189, Hash #2

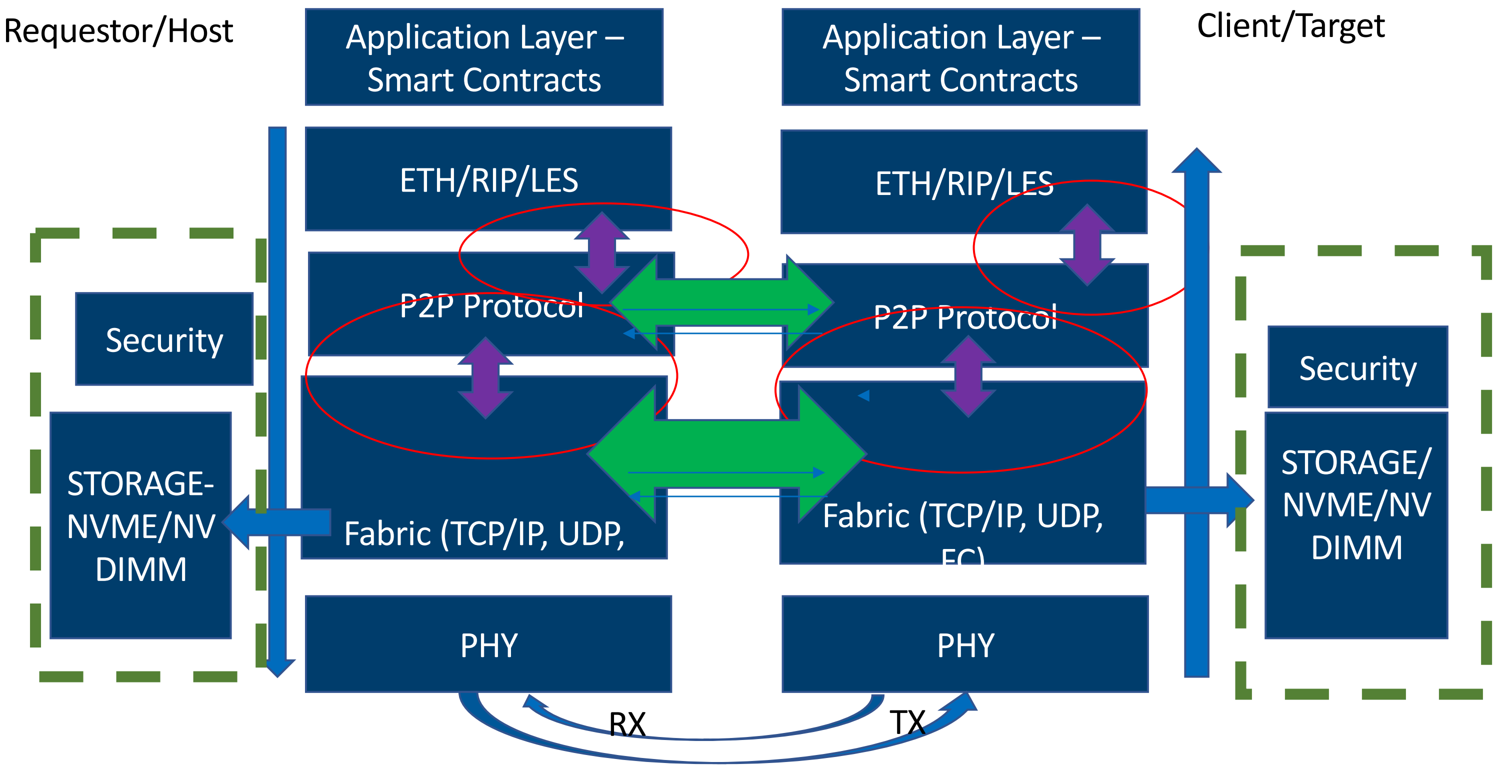
Scoop 4095: Hash #8191, Hash #0

SNIA

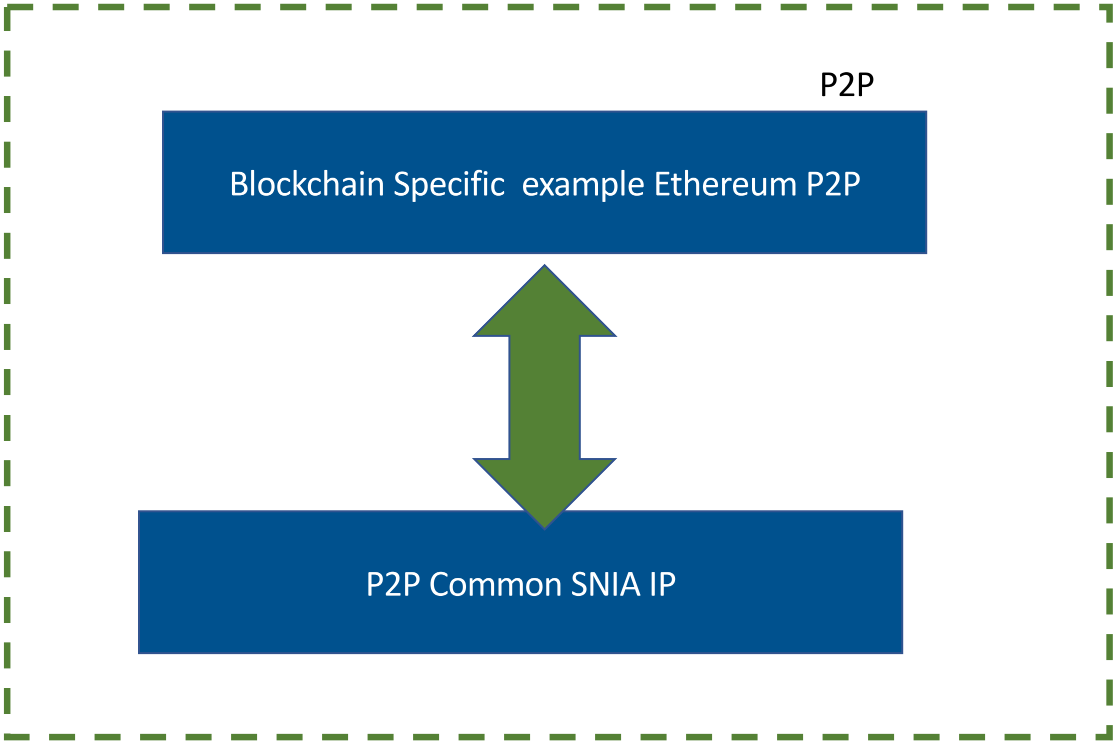
Proposed Blockchain Interoperability Architecture


Distributed Ledger level N
Distributed Ledger level 1
Distributed Ledger level 0
Software Key
Blockchain Service Layer N
Blockchain Service Layer 1
Blockchain Service Layer 0
Smart Control Nodes
Authentication
API
API
FW Blockchain Layer
Blockchain Private Key/Public Key
Storage/Security Block

Integration at protocol level



P2P Communication Interface

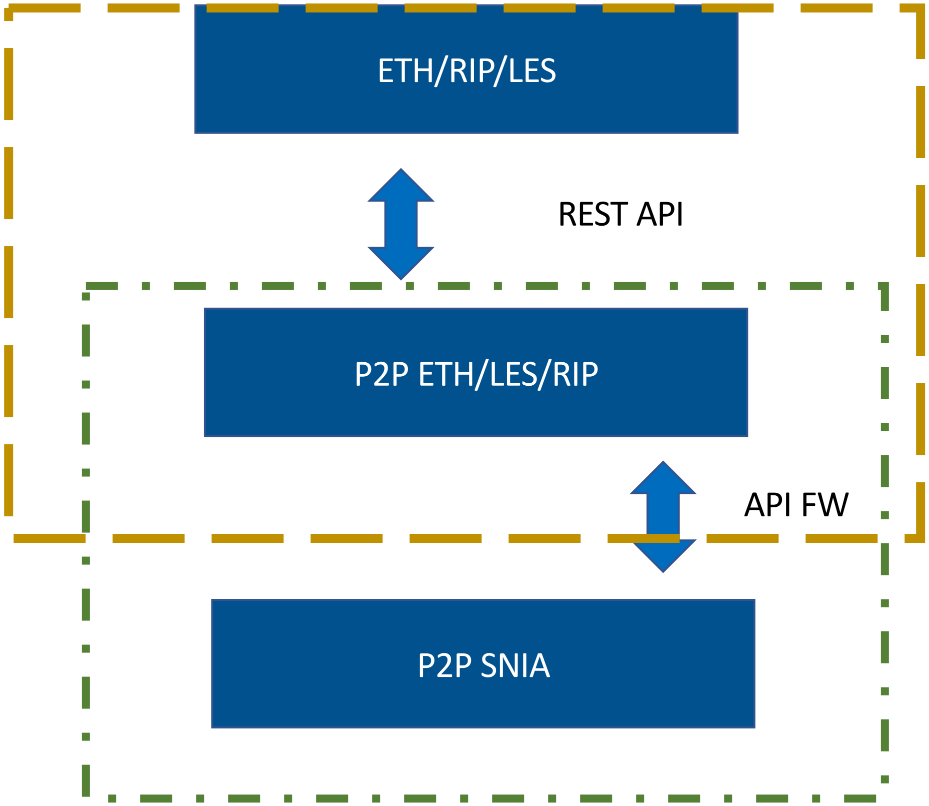


P2P

Blockchain Specific example Ethereum P2P

P2P Common SNIA IP

Extended Overview of the P2P Communication Protocol



ETH/RIP/LES

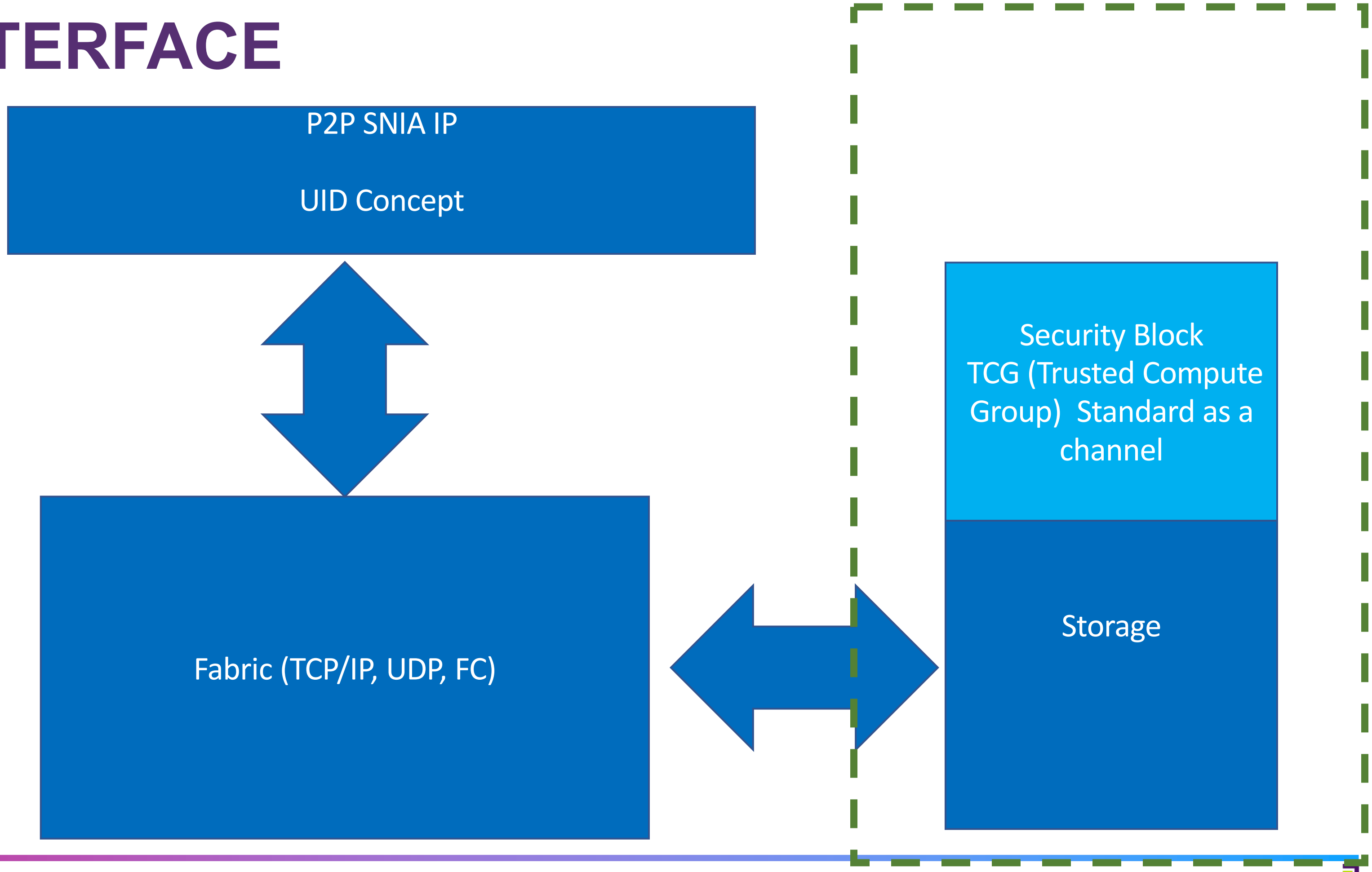
REST API

P2P ETH/LES/RIP

API FW

P2P SNIA

SNIA IP INTERFACE



P2P SNIA IP

UID Concept

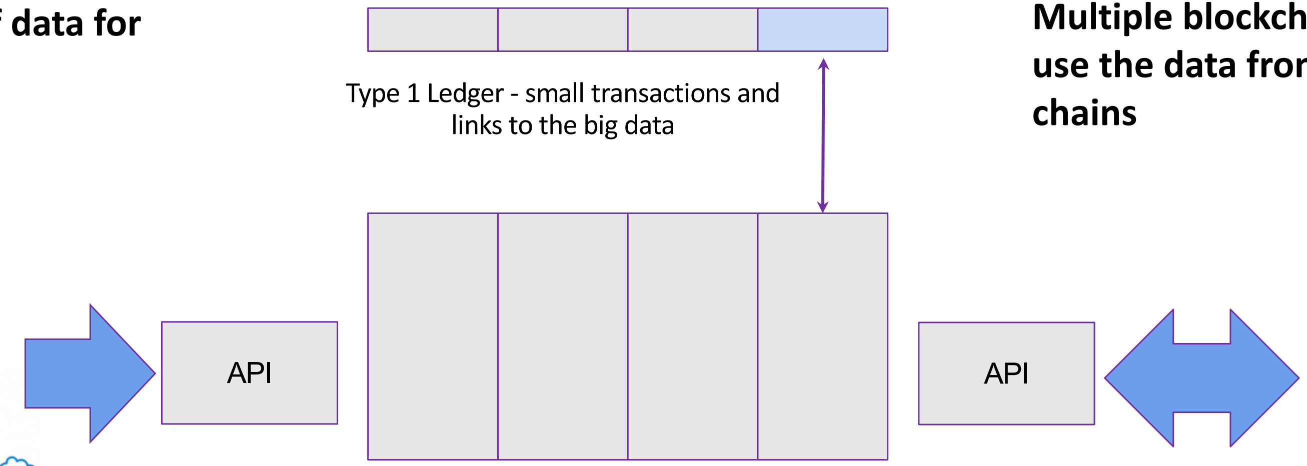
Fabric (TCP/IP, UDP, FC)

Security Block
TCG (Trusted Compute Group) Standard as a channel

Storage

External sources of data for blockchain

Each incoming signal contains SLA, the name of blockchain it has to be processed with and the name of master chain (optional)


Proto-chains (parachains)
Type 1 Ledger - small transactions and links to the big data
API
API

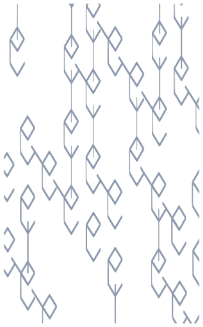
Type 2 Ledger - big data (stateless)

say have multiple chains of type 1 and of type 2

Say support

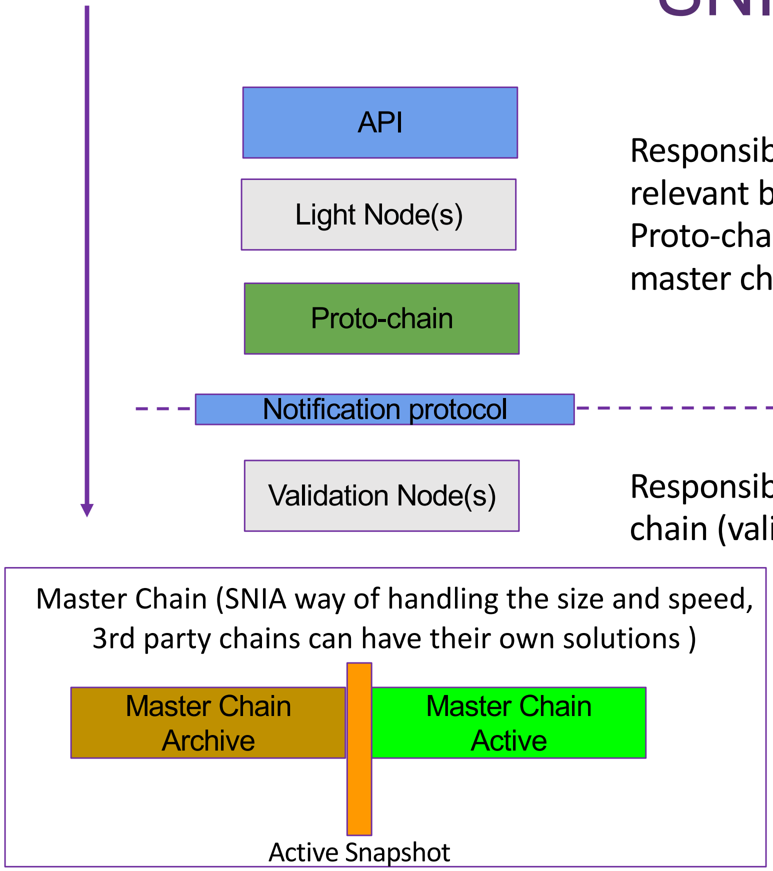
- API for adding and reading chains
- Spread protocol (way to create multiple copies of a chain inside of our network)
- Messaging protocol
- Signaling protocol

Multiple blockchains that use the data from type 1/2 chains



Hyperledger, Etherium, Polkadot, Stellar, etc.

SNIA Proto-Chain

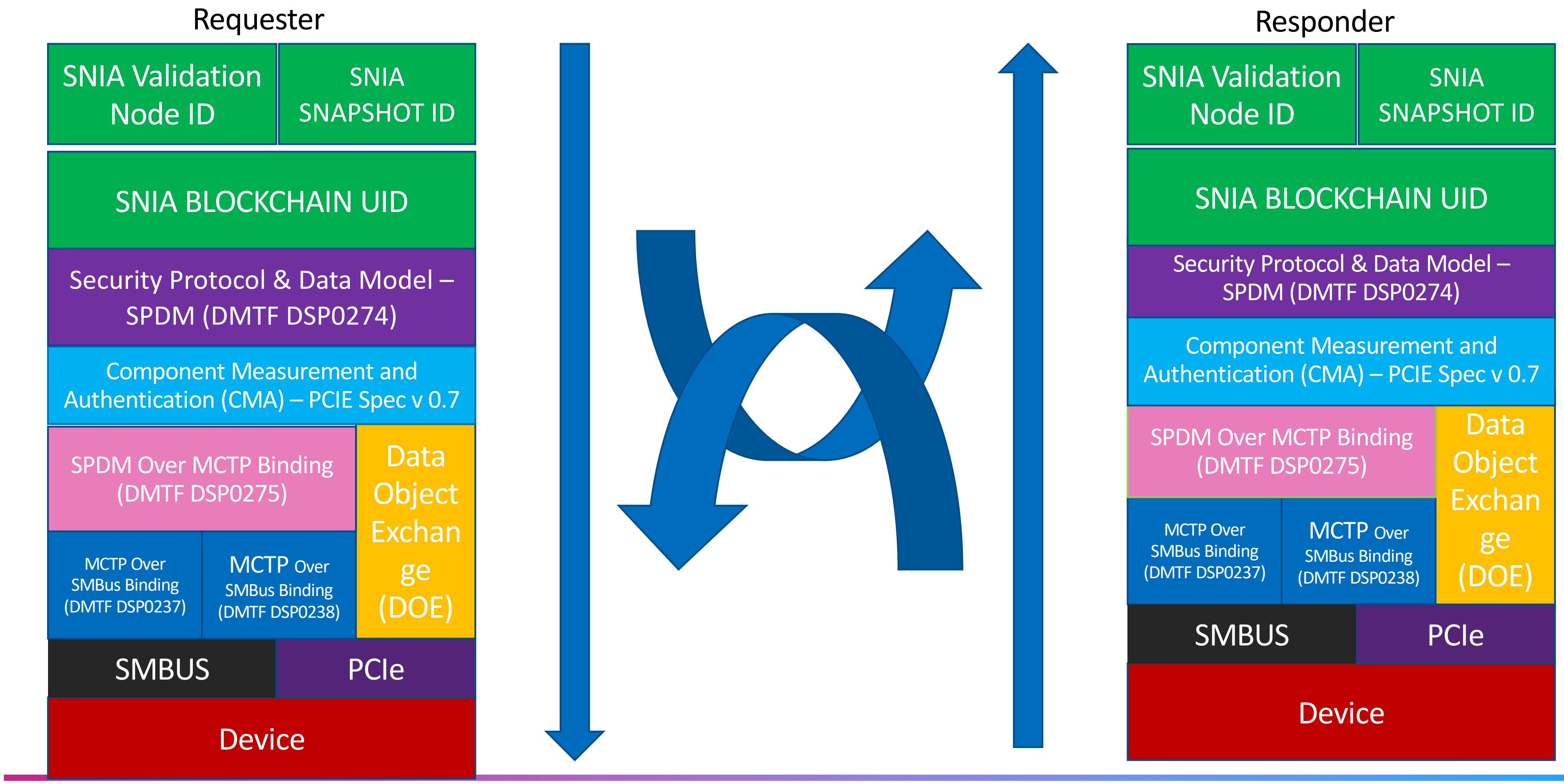


Responsible for creation of proto-chains according to SLA and for notification of relevant blockchain (or validation node)
Proto-chain is non-validated chain of events, data and transactions that can join master chain after their validation

Responsible for validation of blocks in Proto-chain based on the data in a master chain (validation and consensus depend on the logic of the master chain)

SNIA Master chain that consists three parts - active growing part of the chain, snapshot of all the history (actual states) from archive part and the archive part of the chain

Protocol Low Level Communication



Requester

SNIA Validation Node ID | SNIA SNAPSHOT ID

SNIA BLOCKCHAIN UID

Security Protocol & Data Model – SPDM (DMTF DSP0274)

Component Measurement and Authentication (CMA) – PCIE Spec v 0.7

SPDM Over MCTP Binding (DMTF DSP0275)

MCTP Over SMBus Binding (DMTF DSP0237) | MCTP Over SMBus Binding (DMTF DSP0238)

Data Object Exchange (DOE)

SMBUS | PCIe

Device

Responder

SNIA Validation Node ID | SNIA SNAPSHOT ID

SNIA BLOCKCHAIN UID

Security Protocol & Data Model – SPDM (DMTF DSP0274)

Component Measurement and Authentication (CMA) – PCIE Spec v 0.7

SPDM Over MCTP Binding (DMTF DSP0275)

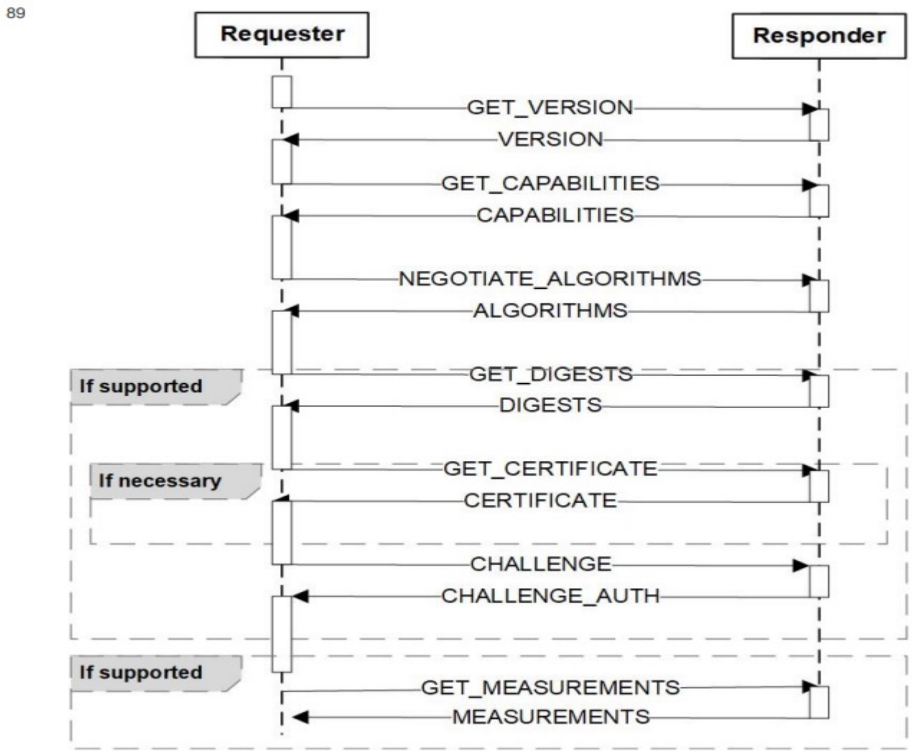
MCTP Over SMBus Binding (DMTF DSP0237) | MCTP Over SMBus Binding (DMTF DSP0238)

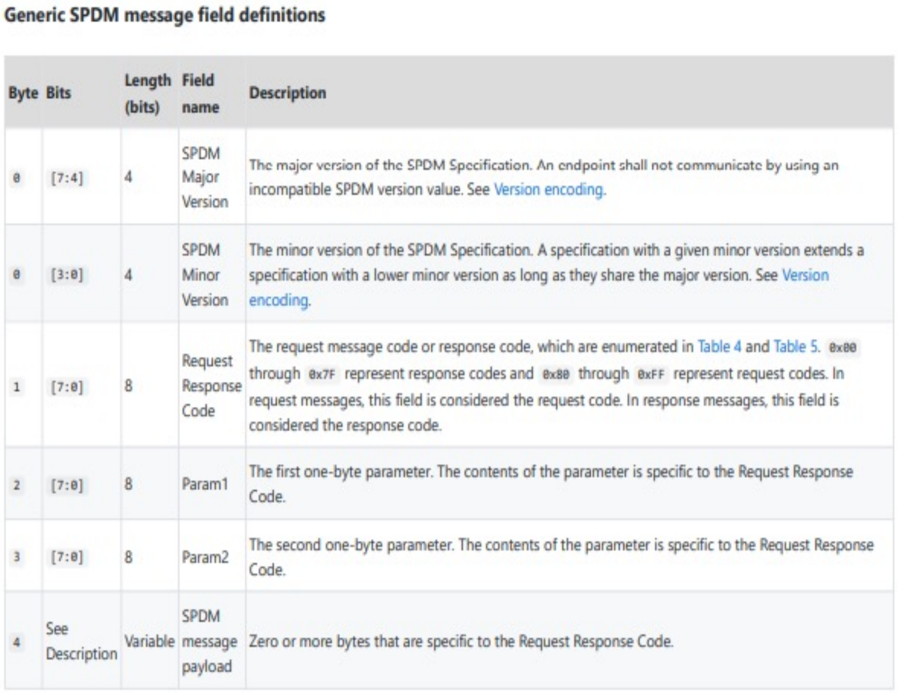
Data Object Exchange (DOE)

SMBUS | PCIe

Device

SPDM Exchange Overview(Security protocol data model)

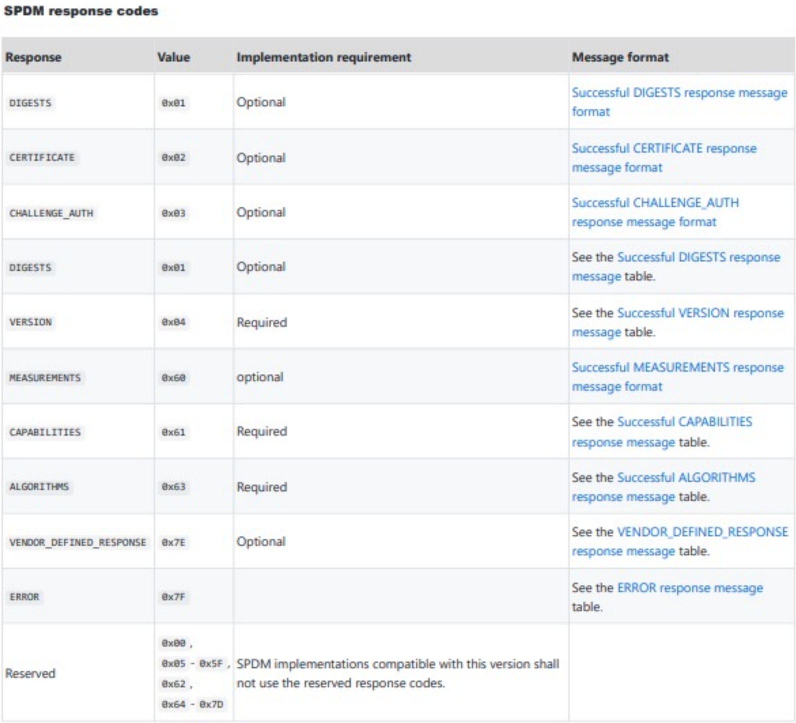




Generic SPDM message field definitions

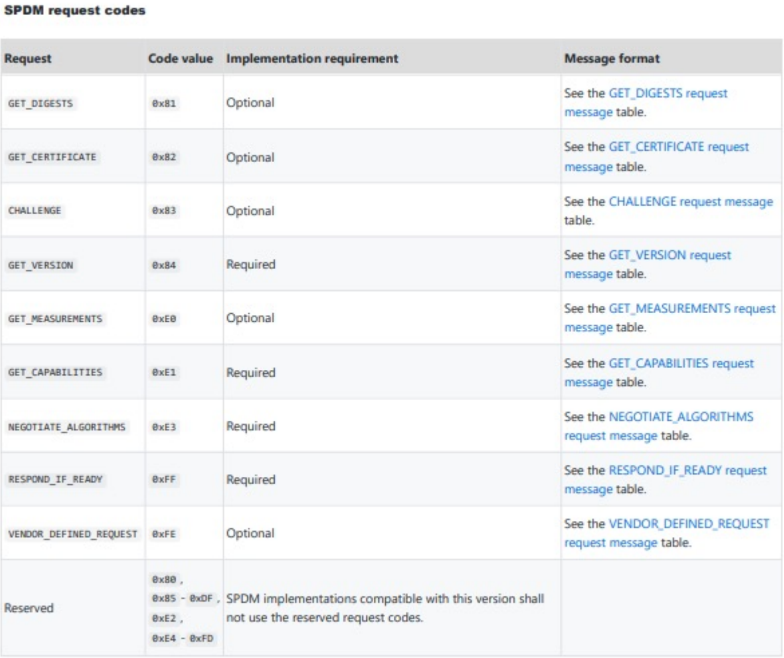
Byte	Bits	Length (bits)	Field name	Description
0	[7:4]	4	SPDM Major Version	The major version of the SPDM Specification. An endpoint shall not communicate by using an incompatible SPDM version value. See Version encoding.
0	[3:0]	4	SPDM Minor Version	The minor version of the SPDM Specification. A specification with a given minor version extends a specification with a lower minor version as long as they share the major version. See Version encoding.
1	[7:0]	8	Request Response Code	The request message code or response code, which are enumerated in Table 4 and Table 5. 0x00 through 0x7F represent response codes and 0x80 through 0xFF represent request codes. In request messages, this field is considered the request code. In response messages, this field is considered the response code.
2	[7:0]	8	Param1	The first one-byte parameter. The contents of the parameter is specific to the Request Response Code.
3	[7:0]	8	Param2	The second one-byte parameter. The contents of the parameter is specific to the Request Response Code.
4	See Description	Variable	SPDM message payload	Zero or more bytes that are specific to the Request Response Code.

SPDM Request and Response Codes



SPDM response codes

Response	Value	Implementation requirement	Message format
`DIGESTS`	`0x01`	Optional	Successful DIGESTS response message format
`CERTIFICATE`	`0x02`	Optional	Successful CERTIFICATE response message format
`CHALLENGE_AUTH`	`0x03`	Optional	Successful CHALLENGE_AUTH response message format
`DIGESTS`	`0x01`	Optional	See the Successful DIGESTS response message table.
`VERSION`	`0x04`	Required	See the Successful VERSION response message table.
`MEASUREMENTS`	`0x60`	optional	Successful MEASUREMENTS response message format
`CAPABILITIES`	`0x61`	Required	See the Successful CAPABILITIES response message table.
`ALGORITHMS`	`0x63`	Required	See the Successful ALGORITHMS response message table.
`VENDOR_DEFINED_RESPONSE`	`0x7E`	Optional	See the VENDOR_DEFINED_RESPONSE response message table.
`ERROR`	`0x7F`		See the ERROR response message table.
Reserved	`0x00` , `0x05` - `0x5F` , `0x62` , `0x64` - `0x7D`	SPDM implementations compatible with this version shall not use the reserved response codes.	

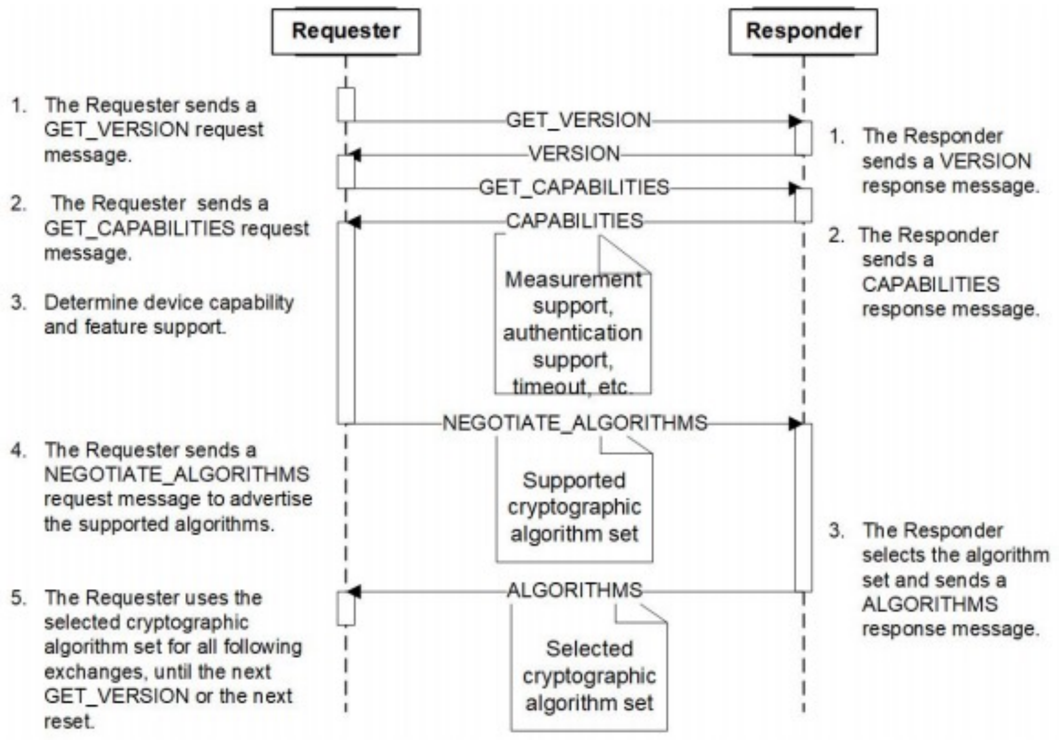


SPDM request codes

Request	Code value	Implementation requirement	Message format
`GET_DIGESTS`	`0x81`	Optional	See the GET_DIGESTS request message table.
`GET_CERTIFICATE`	`0x82`	Optional	See the GET_CERTIFICATE request message table.
`CHALLENGE`	`0x83`	Optional	See the CHALLENGE request message table.
`GET_VERSION`	`0x84`	Required	See the GET_VERSION request message table.
`GET_MEASUREMENTS`	`0xE0`	Optional	See the GET_MEASUREMENTS request message table.
`GET_CAPABILITIES`	`0xE1`	Required	See the GET_CAPABILITIES request message table.
`NEGOTIATE_ALGORITHMS`	`0xE3`	Required	See the NEGOTIATE_ALGORITHMS request message table.
`RESPOND_IF_READY`	`0xFF`	Required	See the RESPOND_IF_READY request message table.
`VENDOR_DEFINED_REQUEST`	`0xFE`	Optional	See the VENDOR_DEFINED_REQUEST request message table.
Reserved	`0x80` , `0x85` - `0xDF` , `0xE2` , `0xE4` - `0xFD`	SPDM implementations compatible with this version shall not use the reserved request codes.	

Capability and Negotiation Flow

Capability discovery and negotiation flow

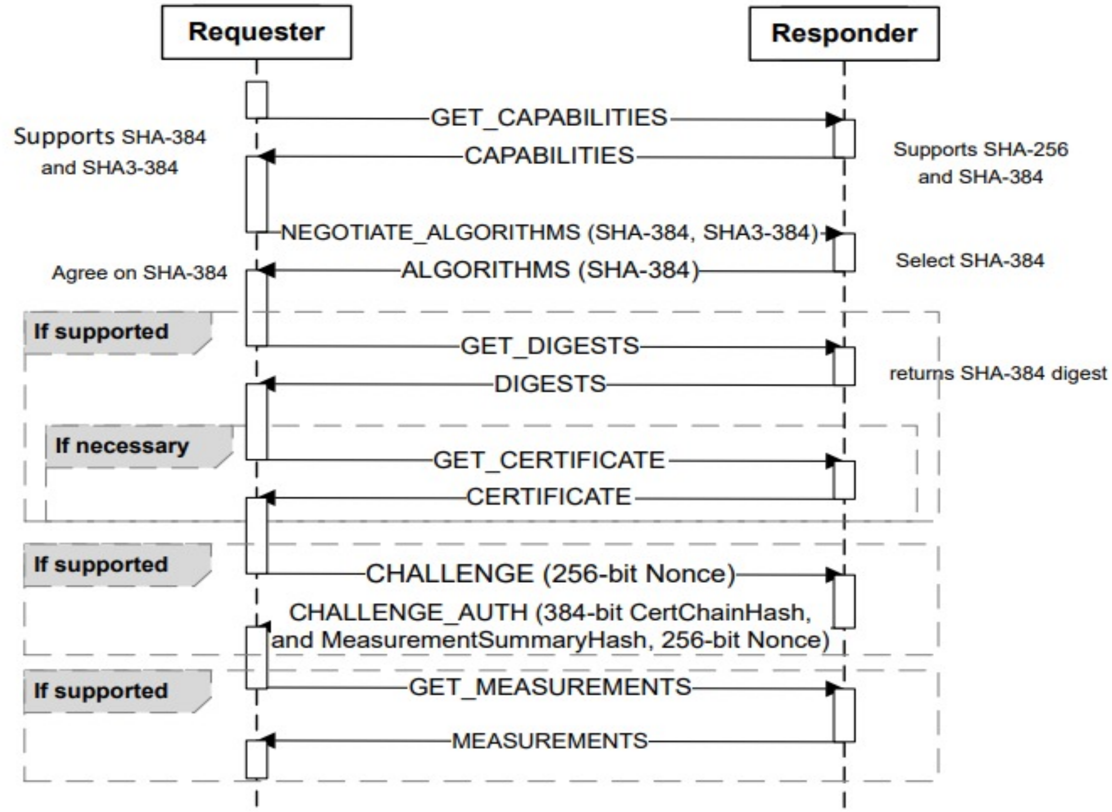

Requester
Responder
GET_VERSION
VERSION
GET_CAPABILITIES
CAPABILITIES
Measurement support, authentication support, timeout, etc.
NEGOTIATE_ALGORITHMS
Supported cryptographic algorithm set
ALGORITHMS
Selected cryptographic algorithm set

1. The Requester sends a GET_VERSION request message.
2. The Requester sends a GET_CAPABILITIES request message.
3. Determine device capability and feature support.
4. The Requester sends a NEGOTIATE_ALGORITHMS request message to advertise the supported algorithms.
5. The Requester uses the selected cryptographic algorithm set for all following exchanges, until the next GET_VERSION or the next reset.

1. The Responder sends a VERSION response message.
2. The Responder sends a CAPABILITIES response message.
3. The Responder selects the algorithm set and sends a ALGORITHMS response message.

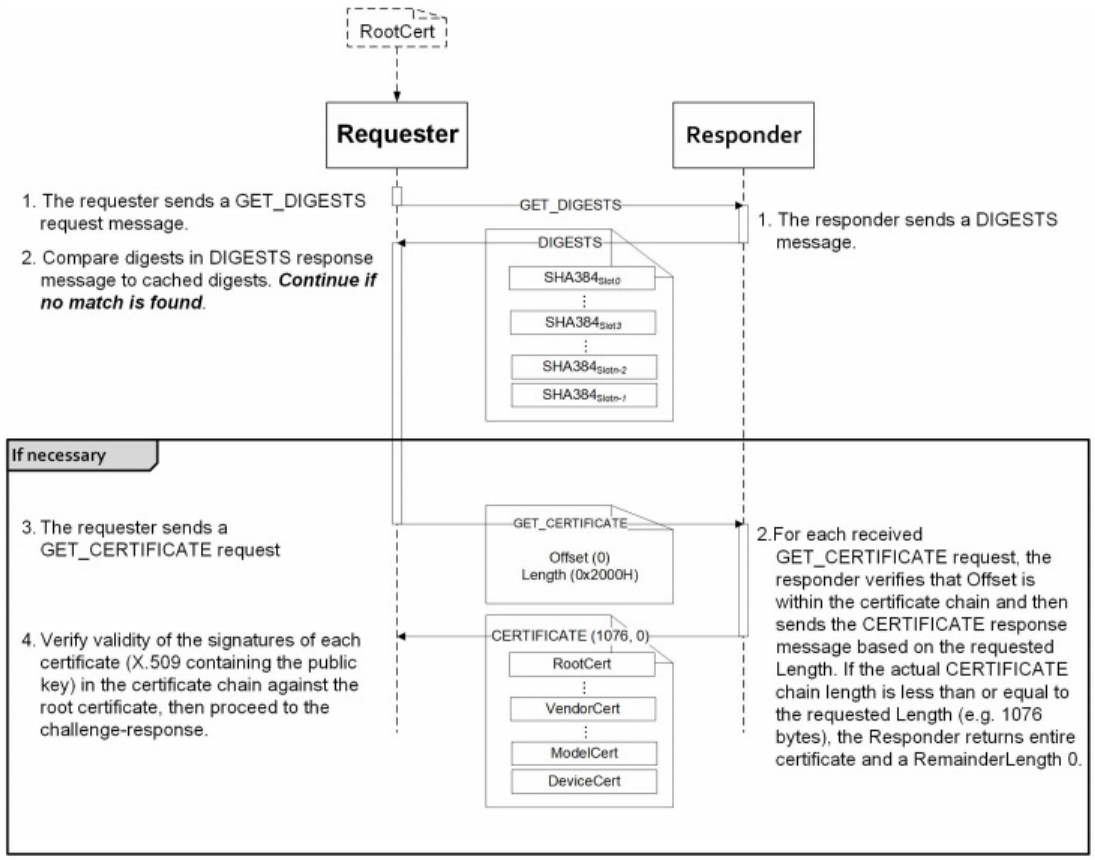
Hashing Algorithm

Hashing algorithm selection: Example 1



Authentication

Responder authentication: Example certificate retrieval flow



1. The requester sends a GET_DIGESTS request message.

2. Compare digests in DIGESTS response message to cached digests. ***Continue if no match is found***.

1. The responder sends a DIGESTS message.

3. The requester sends a GET_CERTIFICATE request

4. Verify validity of the signatures of each certificate (X.509 containing the public key) in the certificate chain against the root certificate, then proceed to the challenge-response.

2. For each received GET_CERTIFICATE request, the responder verifies that Offset is within the certificate chain and then sends the CERTIFICATE response message based on the requested Length. If the actual CERTIFICATE chain length is less than or equal to the requested Length (e.g. 1076 bytes), the Responder returns entire certificate and a RemainderLength 0.

Proposal for SPDM

- SNIA TWG group is working on proposing additional configuration registers to SPDM specification

Communication Between Chains - Handshake

API from Blockchain to SNIA IP Pass

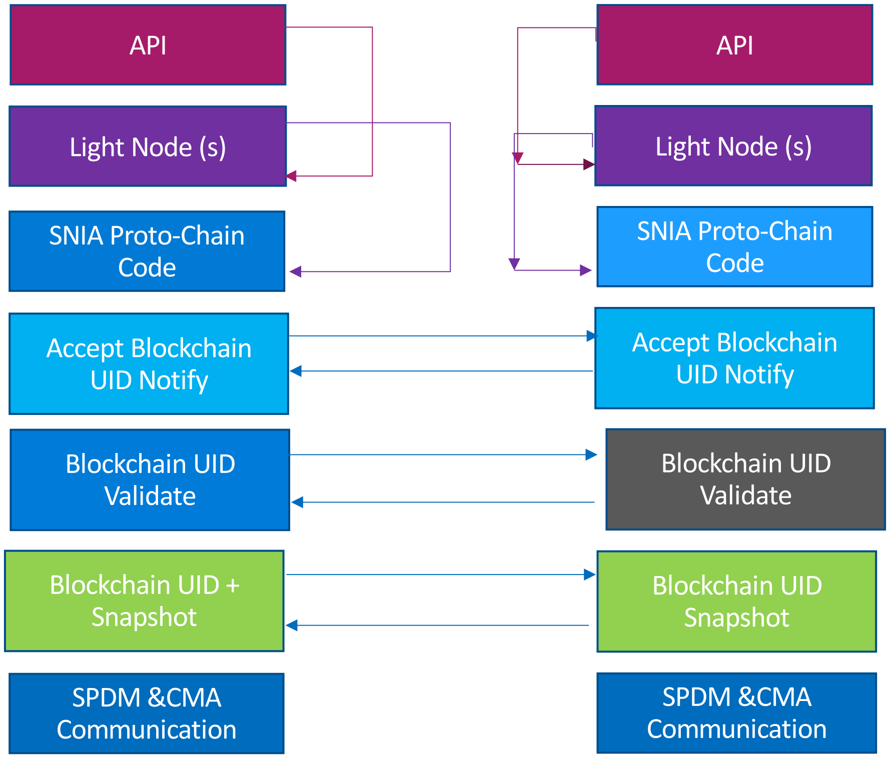
Build a proto-chain to SLA for notification

Requester to Notify about the Blockchain ID

Requester accepts ID as Valid

Requester Snapshop with UID

DMTF, TCG, NVMe, PCIE Protocols



API

Light Node (s)

SNIA Proto-Chain Code

Accept Blockchain UID Notify

Blockchain UID Validate

Blockchain UID + Snapshot

SPDM &CMA Communication

API

Light Node (s)

SNIA Proto-Chain Code

Accept Blockchain UID Notify

Blockchain UID Validate

Blockchain UID Snapshot

SPDM &CMA Communication

API from Blockchain to SNIA IP Pass

Build a proto-chain to SLA for notification

Responder Accepts Blockchain ID

Responder Accepts UID as Valid

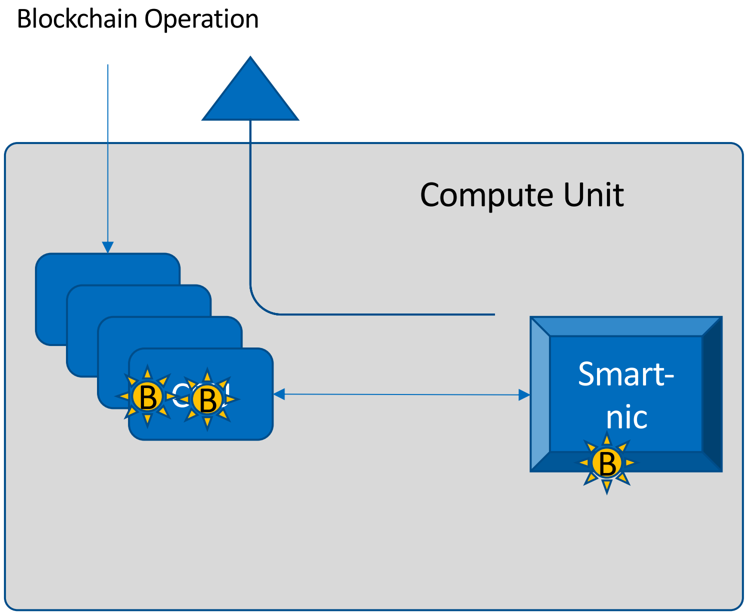
Responder Snapshot with UID

DMTF, TCG, NVMe, PCIE Protocols

Special purpose hardware usages

SmartNIC Architecture

- Management Plane, CLI,REST API, SNMP
- Control Plane, Signaling between network entities
- Data Plane, IPTable, OVS, DPDK, Routing Table
- PCIe, CXL, CCIX, Ethernet, TCP, HTMP

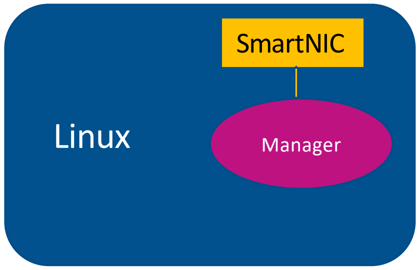


SamartNIC use cases

- CPU is needed for storage, compute and network
- Virtualization has increased recently
- Data path offload
- It adds fraction of CPU overhead and reduces lot of overhead
- Offload network operations to dedicated FPGA
- Can achieve 600 GBPS
- Cut power and cooling consumption

SmartNIC + Blockchain

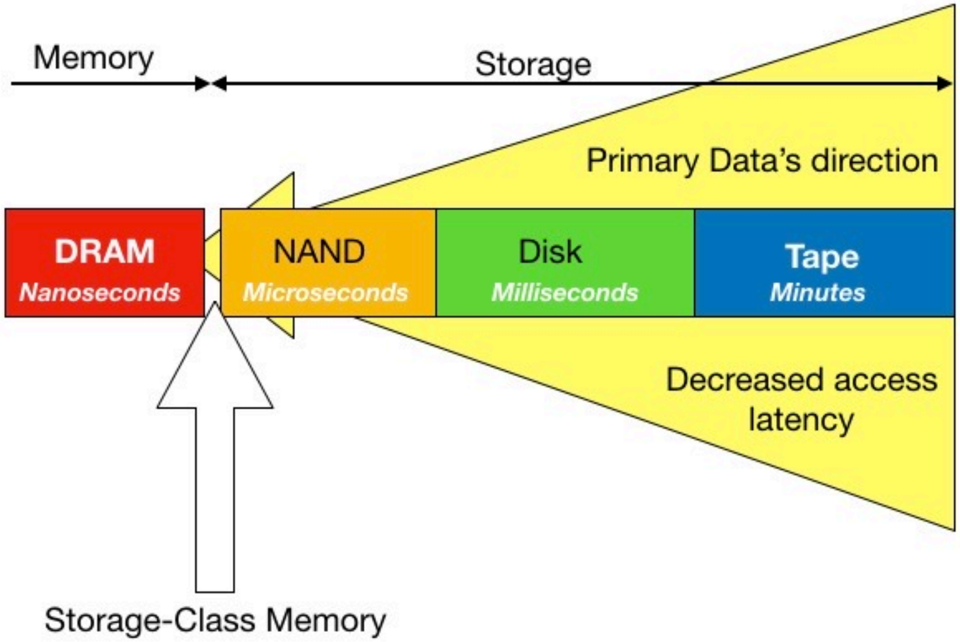
- Develop a glue layer to find out the network intensive task.
- Offload to SmartNIC
- Query for consumption
- Get the result after completion
- Start other operation



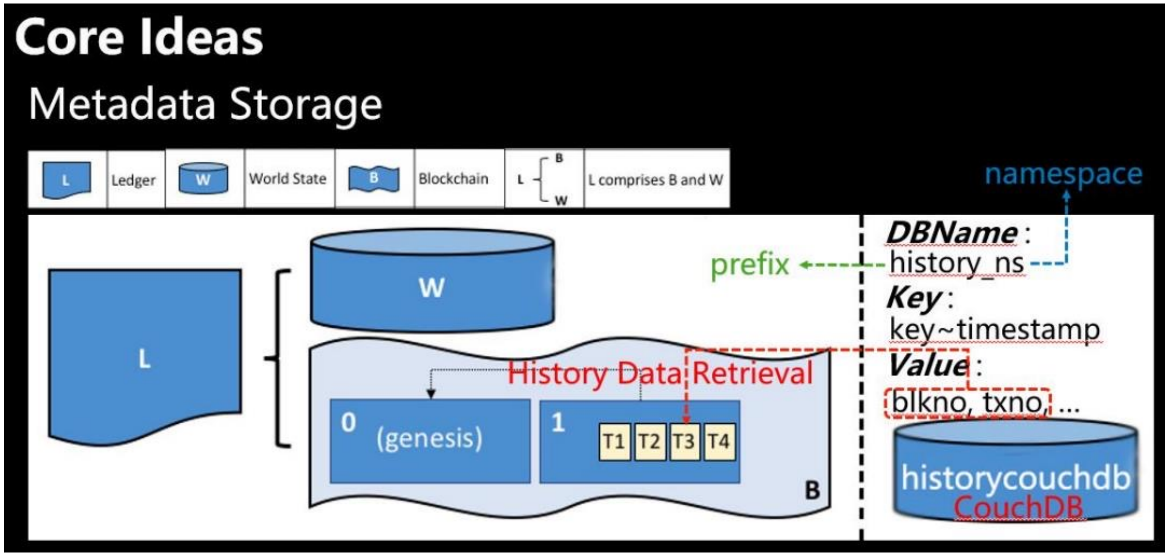
- Systemd interface
- Threads to handle features
- Redfish interface
- RMII-interface (RBT

SCM use cases

- High performance storage tier between SSD and DRAM
- Stores the data after power loss
- Direct filesystem IO by-pass
- Byte addressable



Hyperledger Use Case

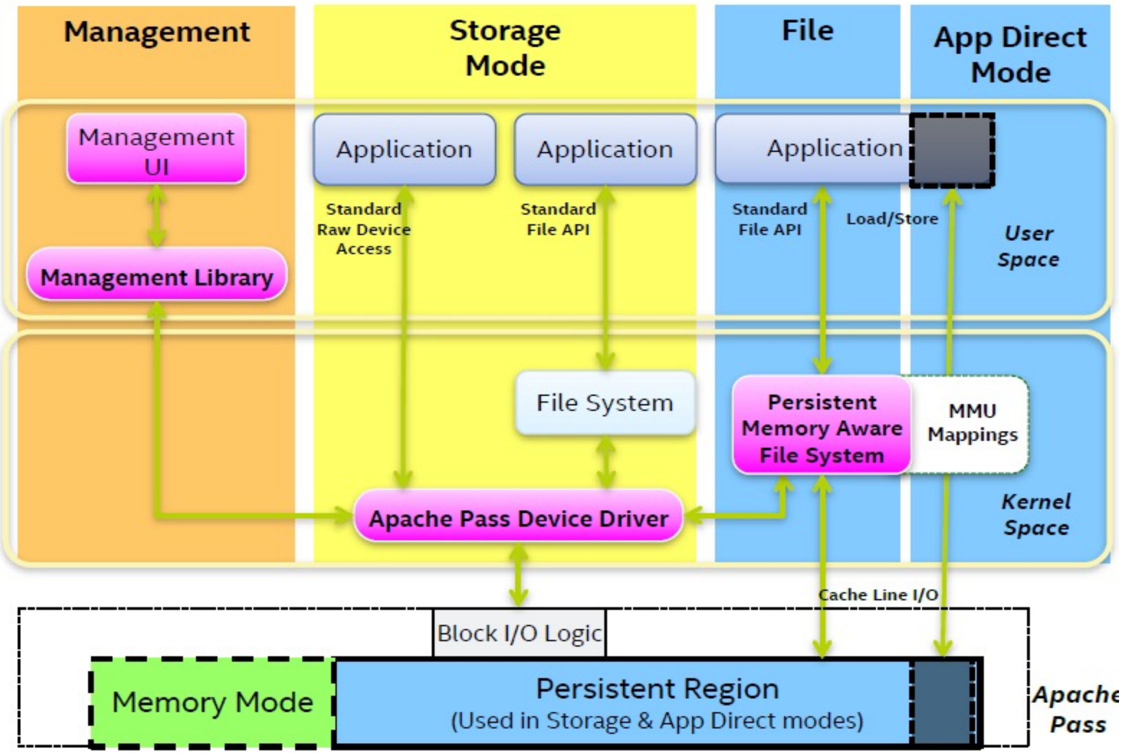


Core Ideas

Metadata Storage

- Off chain requires database to store data
- Normally these are in CouchDB
- Verification and monitory time depends on access time to DB
- Network performance will be slower if we have more access time

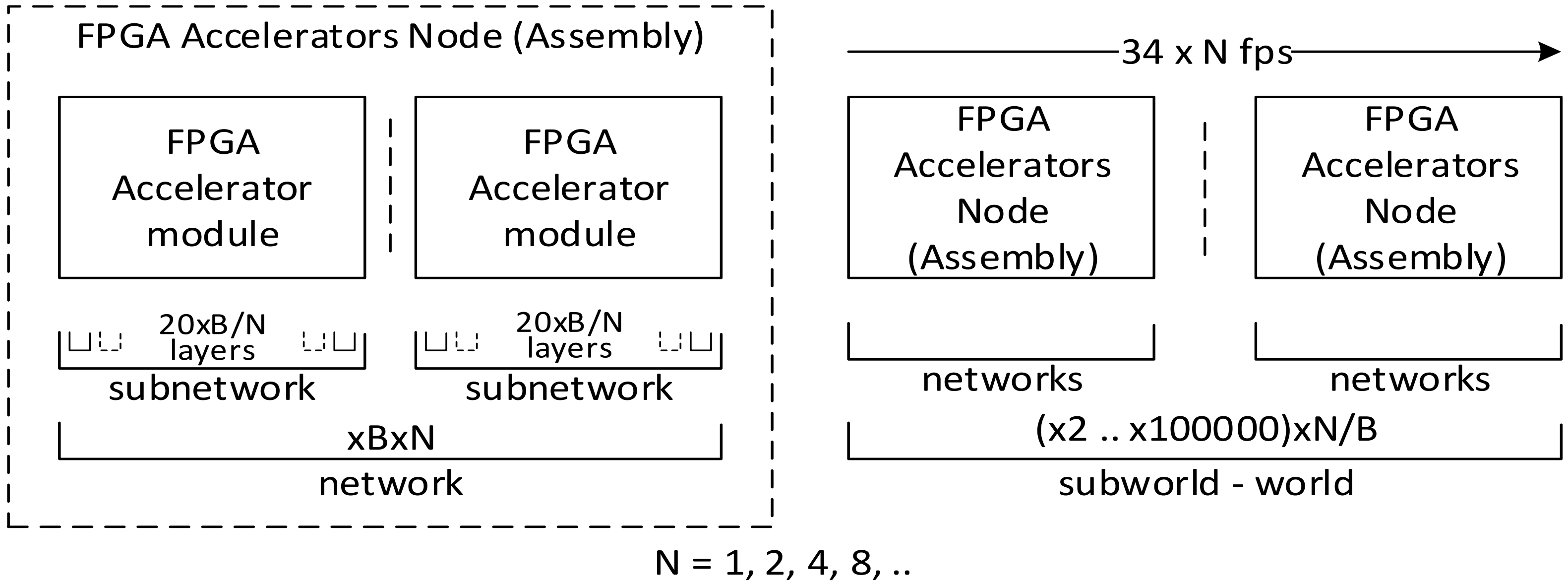
Moving the CouchDB to SCM



- Configure SCM in App-Direct mode
- Set the location of the persistence storage to app-direct location
- OS will take care of access or IO

Neural Networks using FPGA and NVDIMM

Extensibility and Frame rate increase strategy



FPGA Accelerators Node (Assembly)

FPGA Accelerator module

FPGA Accelerator module

20xB/N layers

20xB/N layers

subnetwork

subnetwork

xBxN

network

34 x N fps

FPGA Accelerators Node (Assembly)

FPGA Accelerators Node (Assembly)

networks

networks

(x2 .. x100000)xN/B

subworld - world

N = 1, 2, 4, 8, ..

https://offthechainminers.com/

In Summary

- Newer technology helping blockchain operation optimization.
- Blockchain technology tightly related with storage.
- Interoperability will open lot of other possibility.

SNIA Blockchain Storage Resources

- Interested in contributing to the SNIA Blockchain Storage Technical Working Group?
- Join our group to build Blockchain Data centric storage specification
- For more information visit ….
  - Blockchain Storage Technical Work Group webpage: https://www.snia.org/blockchain
  - Weekend Watch: Blockchain Storage: https://www.snia.org/educational-library/weekend-watch-blockchain-storage-2021
  - Blockchain Storage SNIAVideo/YouTube playlist: https://www.youtube.com/playlist?list=PLH_ag5Km-YUYytvj6LIZ86xYGstWzzB8Y

Your Feedback is Important

- Please take a few moments to rate and provide comments on this webcast
- This webcast and a copy of the slides will be available in the SNIA Educational Library - https://www.snia.org/educational-library
- A Q&A from this webcast, including answers to questions we couldn’t get to today, will be posted on our blog at https://sniansfblog.org/

SNIA

Thank You