ETH zürich

Lightning Filter: High-Speed Traffic Filtering based on DRKey

Juan A. Garcia-Pardo
Research Scientist at Network Security Group, ETH Zürich
29 July 2021, PANRG

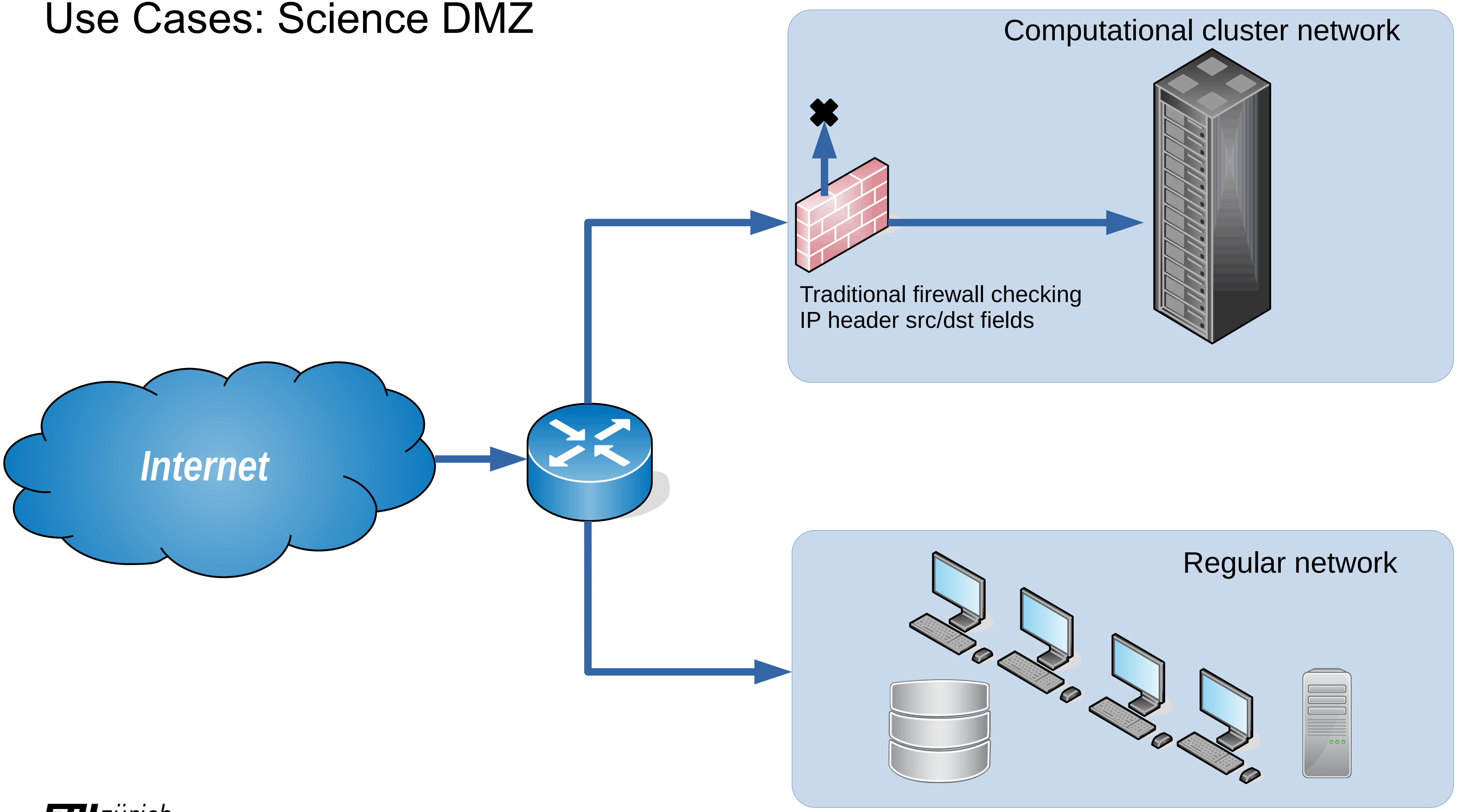
Presentation Index

• Characteristics of Lightning Filter	3
• Use Case (Science DMZ)	4
• Lightning Filter Architecture	5
• Processing Pipeline	6
• How is DRKey used here?	7
• References	10
• Extra (backup slides)	11

Lightning Filter Characteristics

- Based on DRKey
- Suitable for high-speed connections: at the moment up to 160Gbps
- Commodity hardware: dual-socket PC with 4x 40Gbps NICs (<10,000$)
- Every packet is source authenticated cryptographically
- Independence on number of senders or flows

Use Cases: Science DMZ

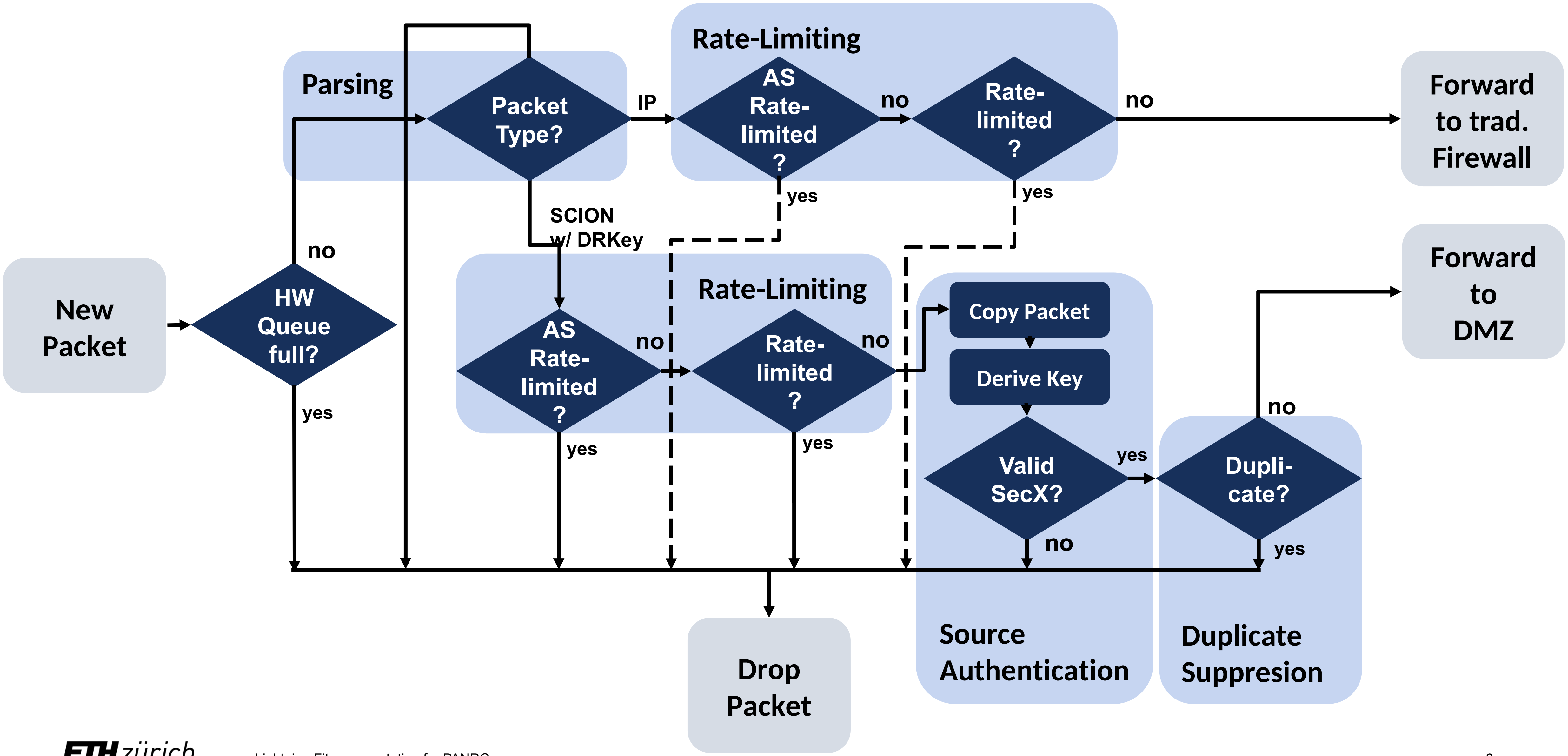


Architecture of the Lightning Filter

- Data Plane
  - Constant time per packet (line rate)

- Control Plane:
  - Fetches DRKeys.
  - Exports metrics.
  - Interacts with the network administrator (rate, blacklist and whitelist settings).

Pipeline (simplified)



How is DRKey used in the Lightning Filter?

- DRKey is necessary for the source authentication of each packet.
- Source authentication is necessary in turn for the duplicate suppression (otherwise trivial to modify bits on packets and resend).
- Source authentication of each packet requires to obtain the keys in nanoseconds. DRKey can accomplish this in two different configurations.
  - With the use of exchanged and pre-fetched L1 DRKeys. When the number of source ASes is small (< 100) and known to not grow large.
  - With the use of a trusted secret value (level 0) for the specific protocol. This is useful if the number of ASes is large or it is desired to let it grow unbound.
- In every DRKey configuration, the side with the Lightning Filter will be on the *DRKey fast path* (fast derivation of the key), while the clients need to interact with their key server (*slow path*).

How is DRKey used in the Lightning Filter?

Level 0

- Called Secret Value. Obtained from a master secret and validity period.
  - The only secret used to derive all other keys.
  - Kept inside the key server. Recreated for each validity period (e.g. 24 hours).

Level 1

(to another AS)

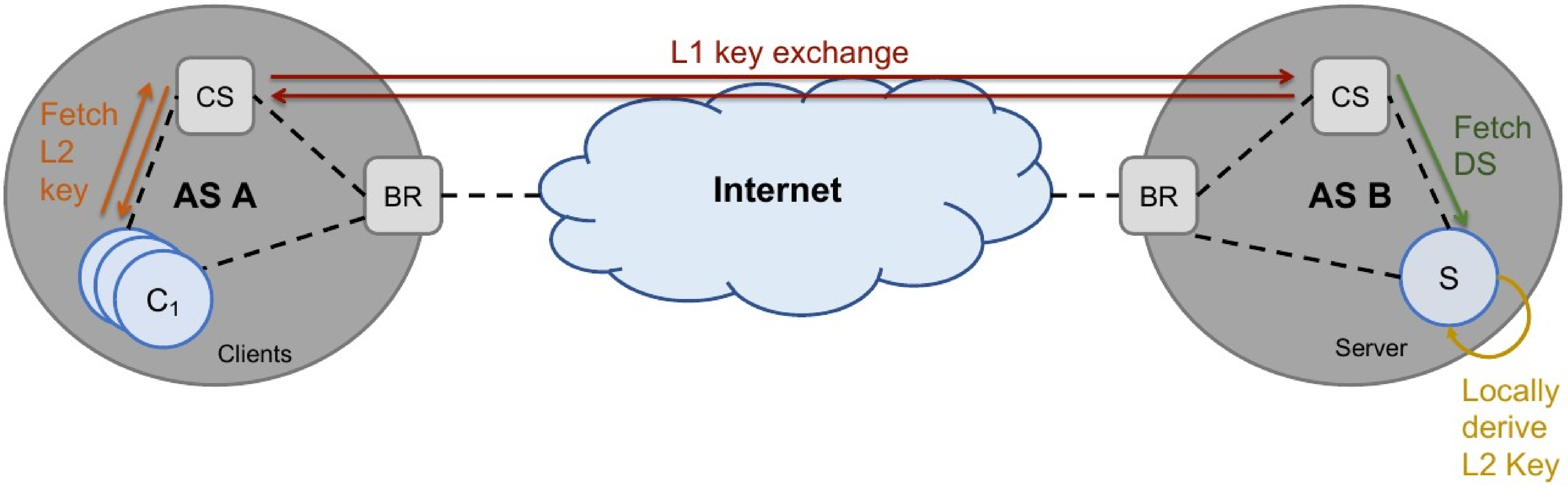
- Obtained from a level 0 key and a destination AS.
  - May include a protocol (a string locking its use for a specific purpose, e.g. “DMZ”).
  - Exchanged with other key servers.

Level 2

(from one entity to another entity for a protocol)

- Workhorse key. Used to authenticate packets.
  - Derived from level 1 key. Must contain a protocol.
  - Key servers always derive it. End hosts derive it or obtain it from their key server.
  - Can be used between two hosts, or the shareholder AS and a host or another AS.

How is DRKey used in the Lightning Filter?
L1 Key Exchange (per AS)



Slow side

Fast side

References

- DRKey paper:

  https://netsec.ethz.ch/publications/papers/piskes_final.pdf

- DRKey I-D (July 2021):

  https://datatracker.ietf.org/doc/html/draft-garciapardo-panrg-drkey-01

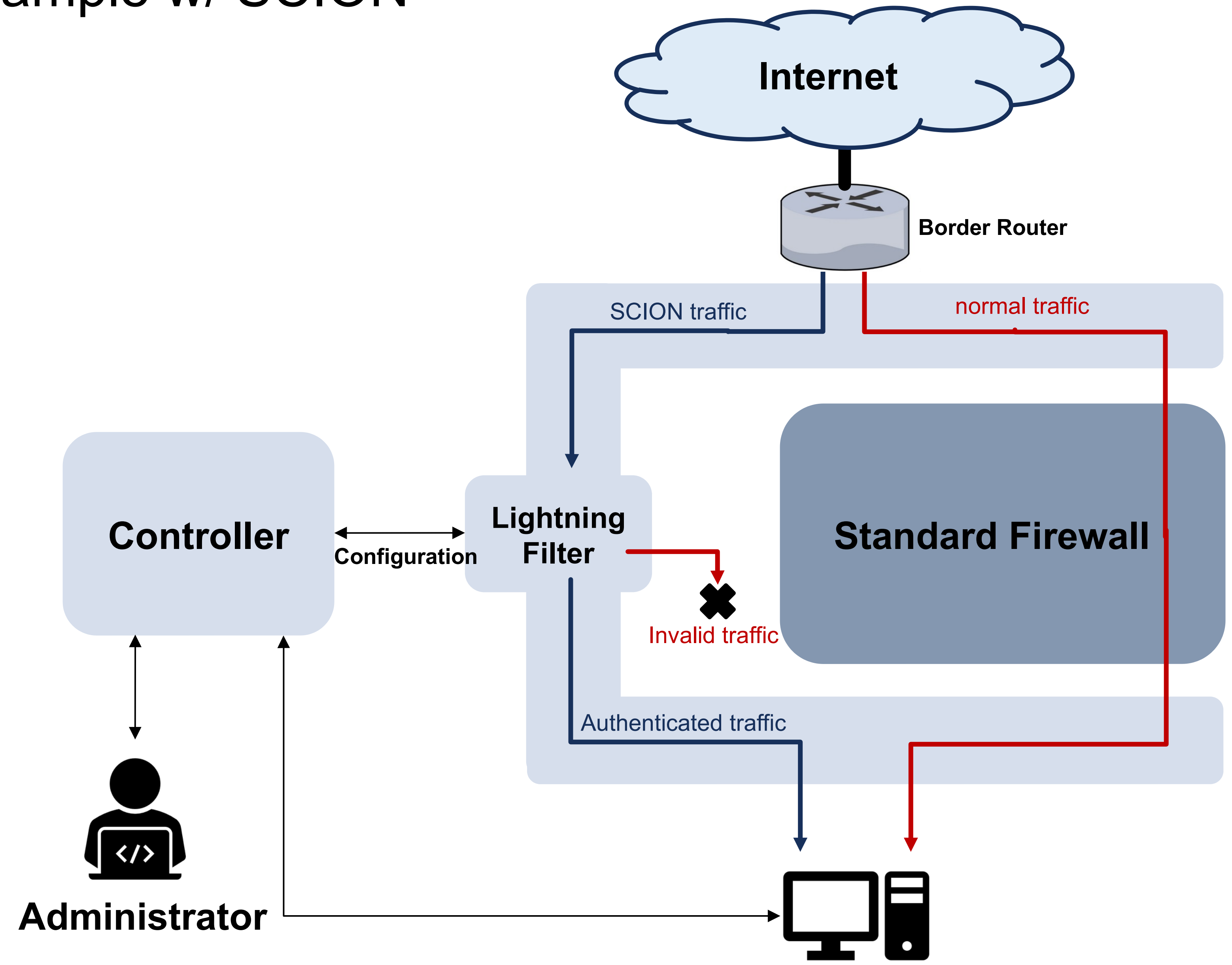
- Lightning Filter implementation:

  https://github.com/netsec-ethz/lightning-filter

BACKUP SLIDES

(use when time permits / to answer questions)

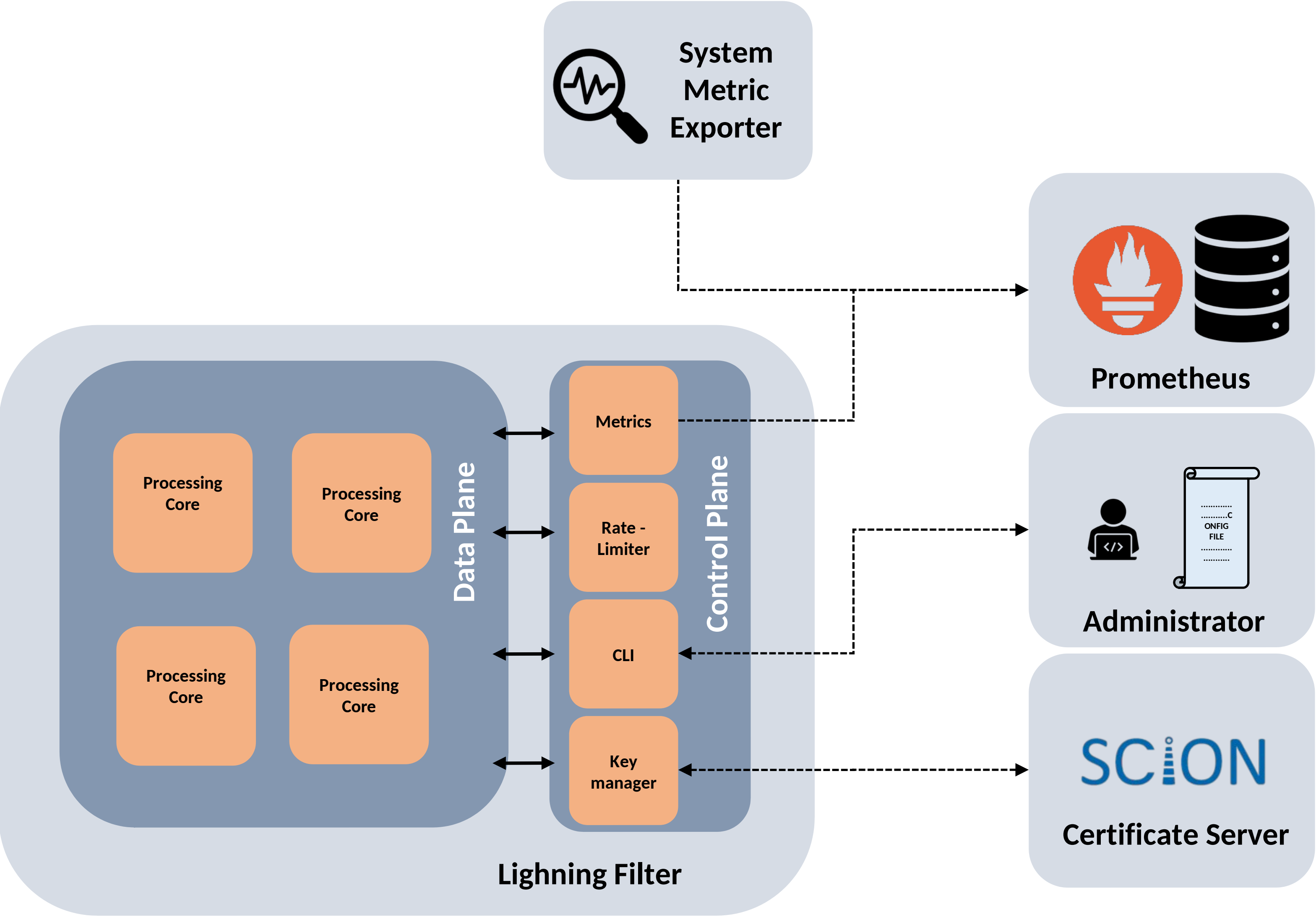
Deployment Example w/ SCION (simplified)



Processing Pipeline

- All components are modules, and can be rearranged into a different pipeline.
- The data plane is time critical, and thus, everything is optimized for it. The interaction between control and data planes is lock-less for efficiency.
- Control plane fetches the L1 DRKeys for the configured ASes, and interacts with the configuration or network administrator to store the traffic rates per source, as well as the blacklist and whitelist of sources.
- The data plane will use the L1 DRKeys to derive L2 per end-host on the fly (nanoseconds) to perform the source authentication.
- If global time synchronization is not available, the timestamp-base filter is off.
- The duplicate suppression module uses bloom filters to efficiently (probabilistically) detect duplicates. The filters are rotated (recreated) periodically.

Architecture of the Lightning Filter



Use Cases: Science DMZ

- Only some machines (the *valid* sources) from some institutions are allowed to access the high performance server cluster.
- There might be hundreds of valid sources, each with hundreds of flows at a given time.
- The computational cluster needs as much network speed as possible.
- The communication between cluster and valid sources is usually carried over the public internet.

- We want to protect the computational cluster from DoS. Traditionally done with a firewall checking src/dst from IP header.
- But IP source spoofing / replaying packets could still DoS the cluster.

How is DRKey used in the Lightning Filter?

$$\text{Secret Value} = SV_A = PBKDF2(\text{validity}, \text{salt}, 1000iter, \text{SHA256})$$

$$\text{Level 1 Key}_{\text{shareholder}=A,\text{other}=B} \equiv K_{A\to B} = \text{PRF}_{SV_A}(B)$$

$$\text{Level 2 Key}^{\text{protocol}} \equiv K^{protocol}_{A:h1\to B:h2} = \text{PRF}_{K_{A\to B}}(\text{"protocol"}, h1, h2)$$

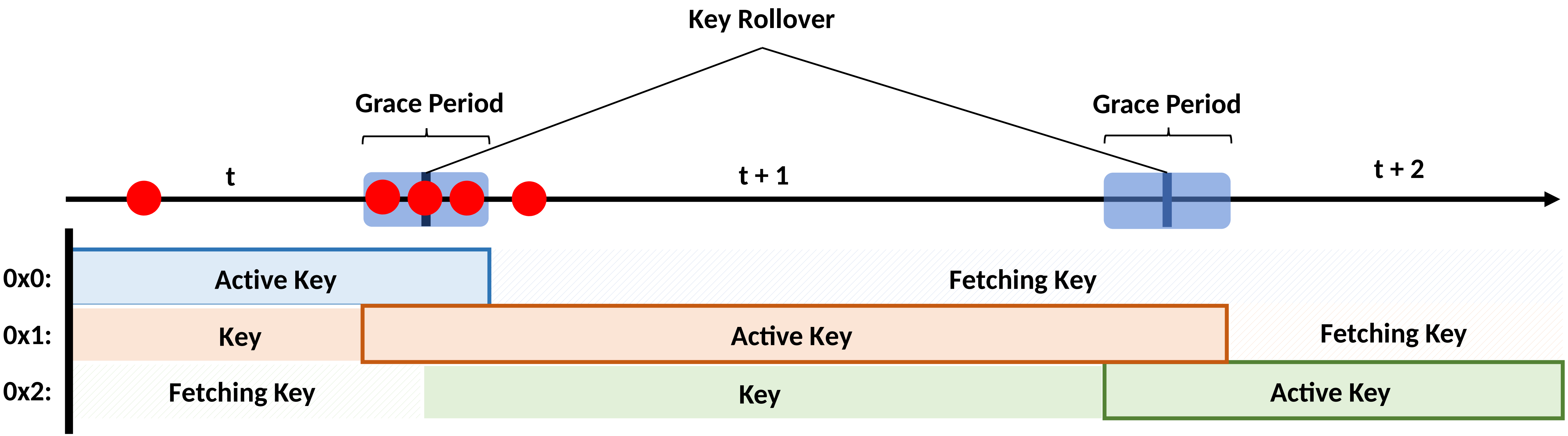
Other possible derivations (configuration II):

$$\text{Protocol Specific Secret Value} \equiv SV_A^{proto} = \text{PRF}_{SV_A}(\text{"}proto\text{"})$$

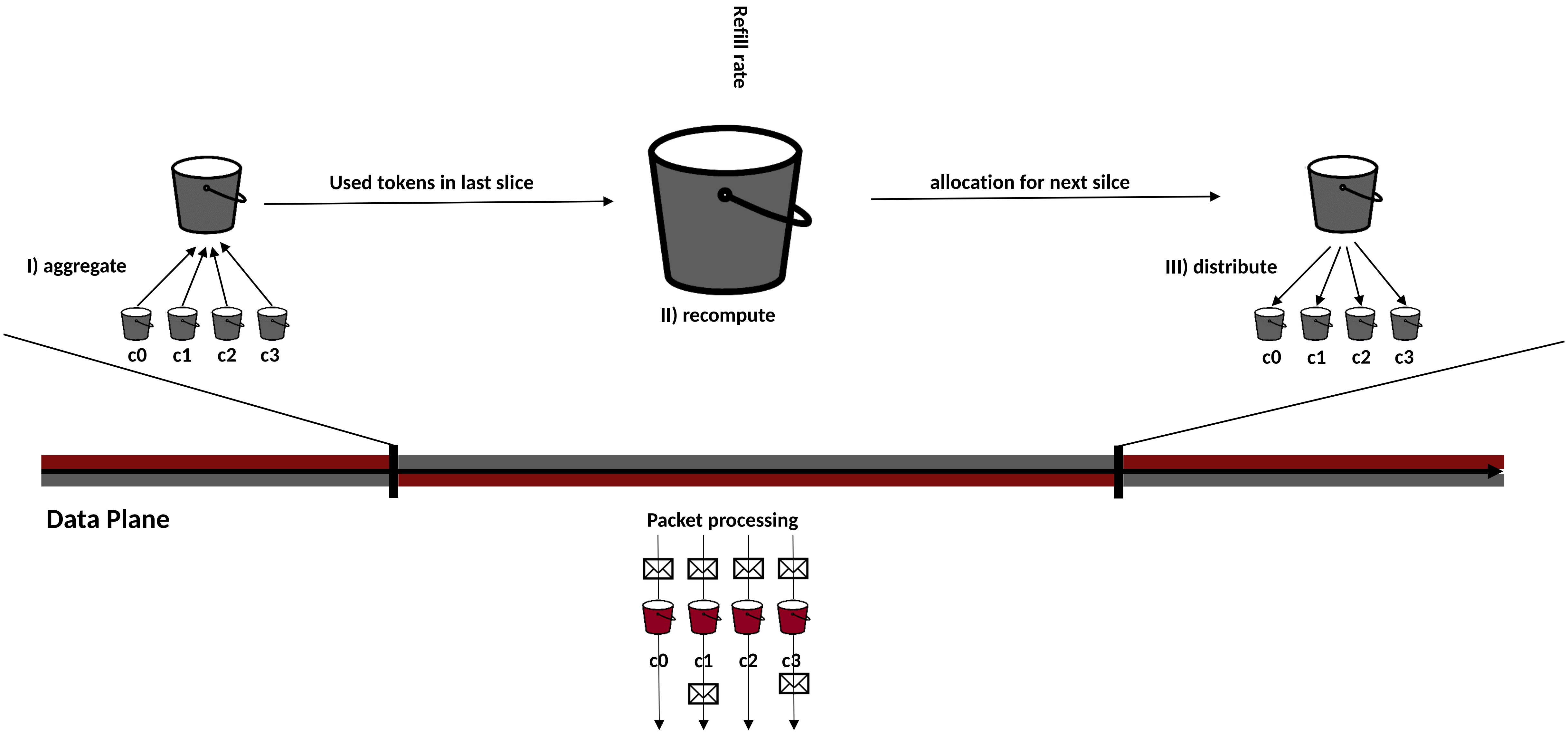
$$\text{Protocol Specific Level 1} \equiv \tilde{K}^{proto}_{A\to B} = \text{PRF}_{SV_A^{proto}}(B)$$

$$\text{Protocol Specific Level 2} \equiv \tilde{K}^{proto}_{A:h1\to B:h2} = \text{PRF}_{\tilde{K}^{proto}_{A\to B}}(h1, h2)$$

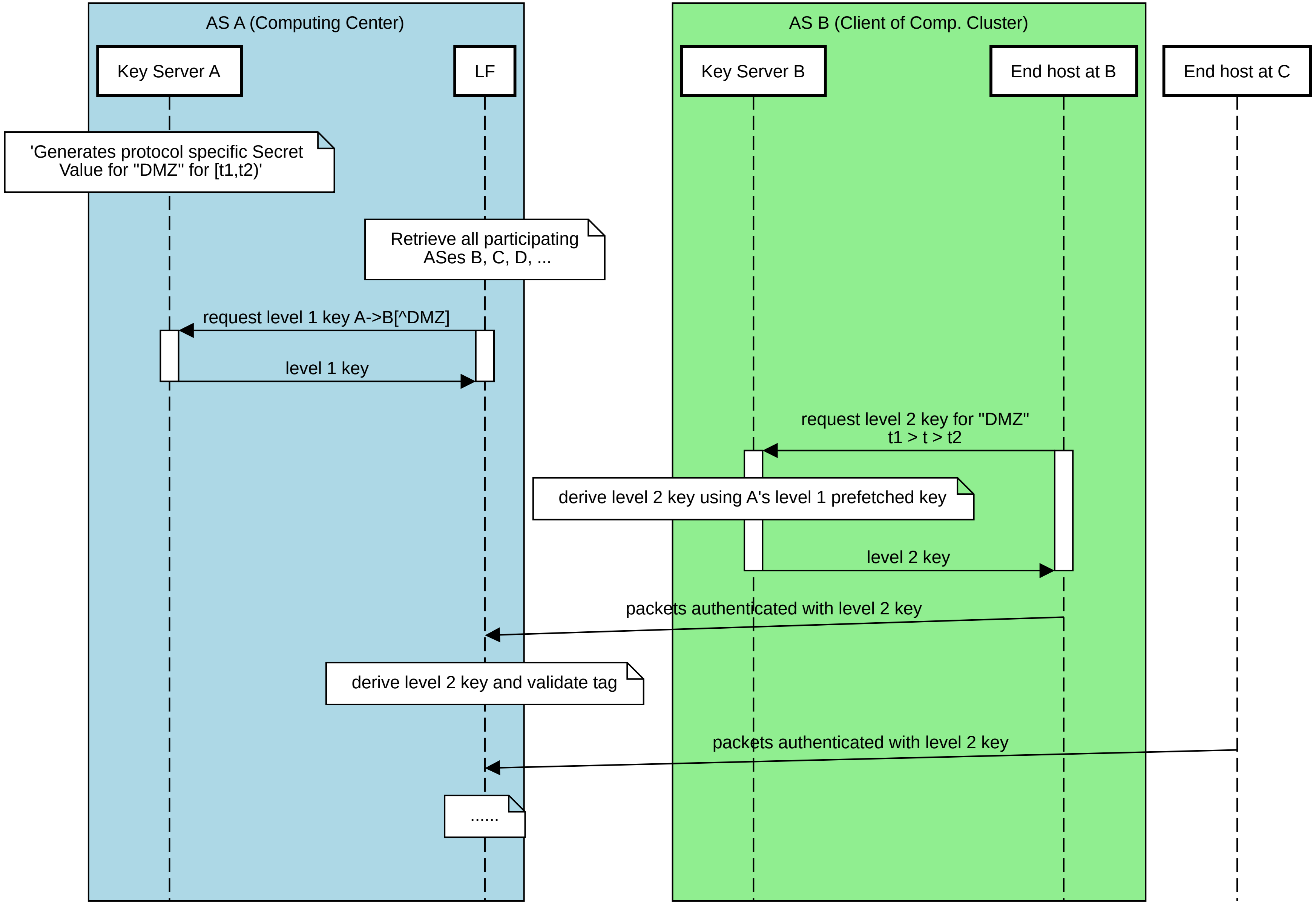
Key Rollover



Rate Limiter



Lightning Filter Interaction



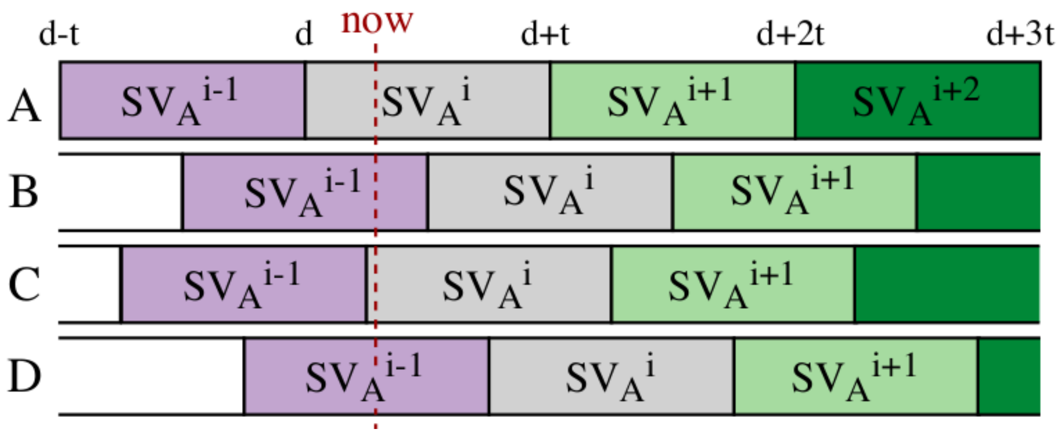
AS A (Computing Center)
AS B (Client of Comp. Cluster)
Key Server A
LF
Key Server B
End host at B
End host at C
'Generates protocol specific Secret Value for "DMZ" for [t1,t2)'
Retrieve all participating ASes B, C, D, ...
request level 1 key A->B[^DMZ]
level 1 key
request level 2 key for "DMZ" t1 > t > t2
derive level 2 key using A's level 1 prefetched key
level 2 key
packets authenticated with level 2 key
derive level 2 key and validate tag
packets authenticated with level 2 key
......

Key Exchange Details

- Because it is typical to have the same validity period (e.g. 24 hours) for many level 1 keys, there could be peaks of level 1 key requests.
- To avoid the concentration, a deterministic function offsetting the validity of the key is used:

$$\mathrm{offset}(A,B) \mapsto [0,t)$$

$$\mathrm{offset}(A,B) = \mathsf{H}(A||B) \mod t$$



- H is a (non cryptographic) hash function.
- The requests are spread uniformly.