Thème :

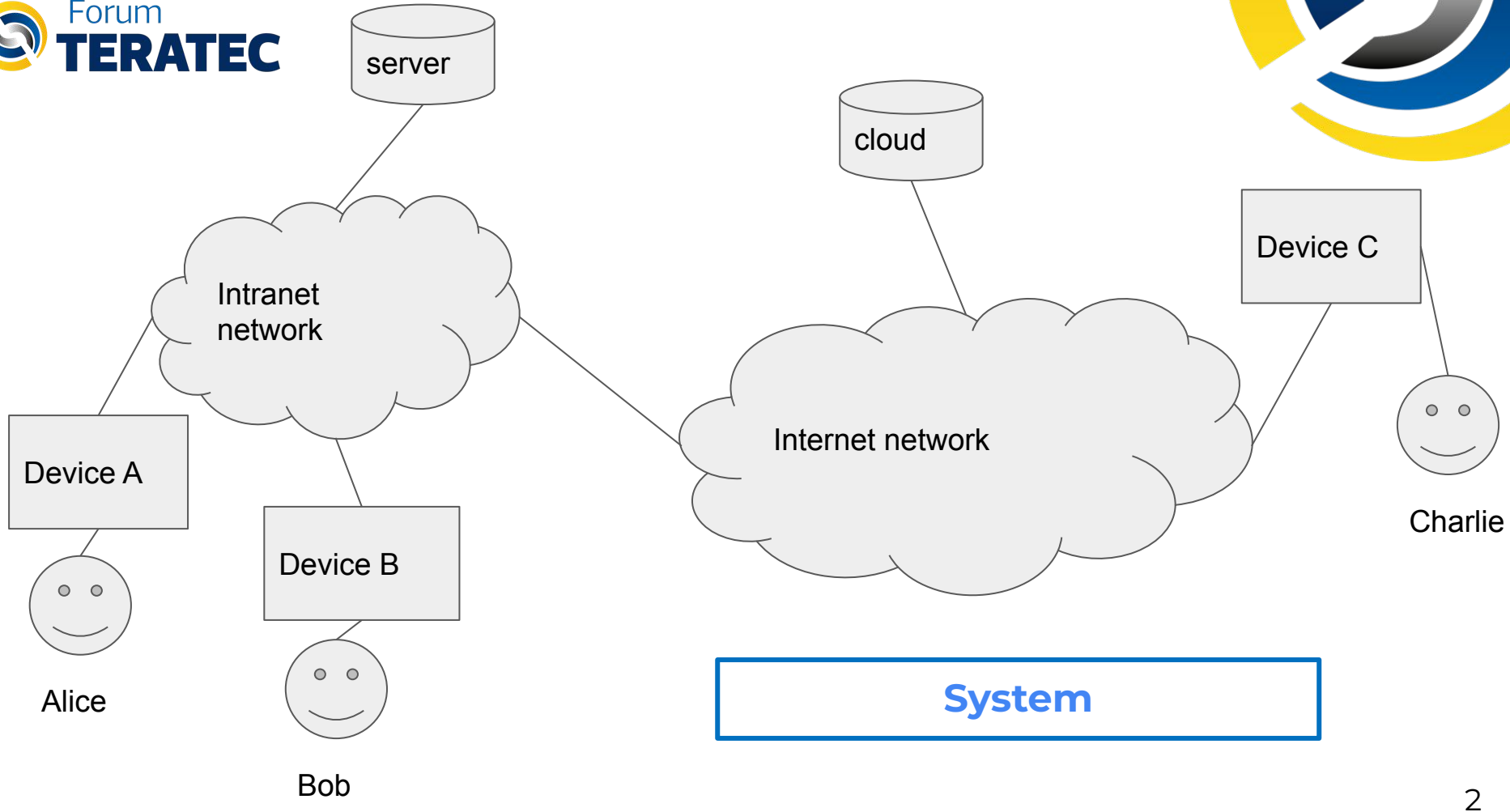
Cipher Data Centric Security

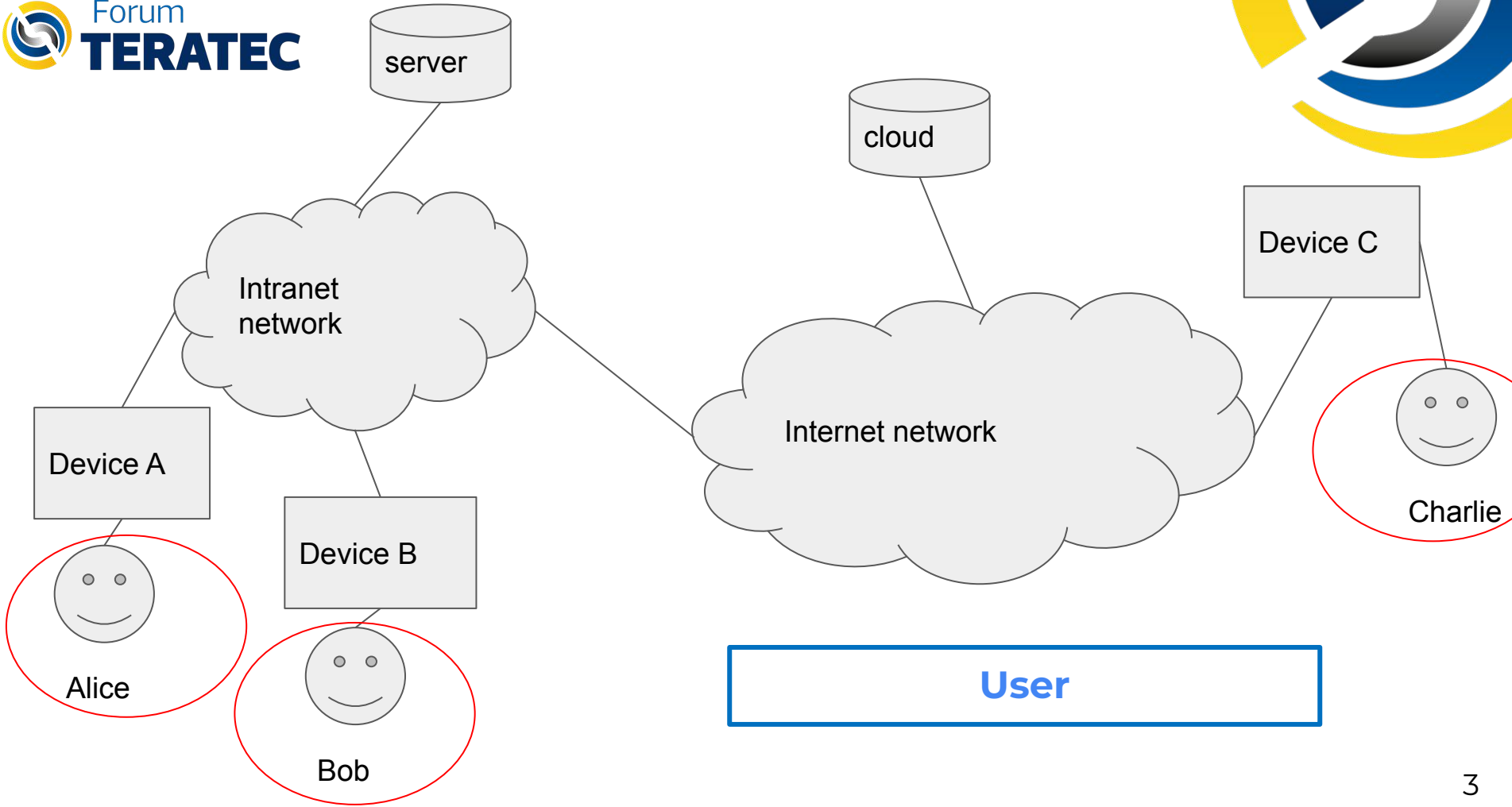
Présenté par :
Thierry Leblond, CEO & Co-fondateur
SCILLE / PARSEC

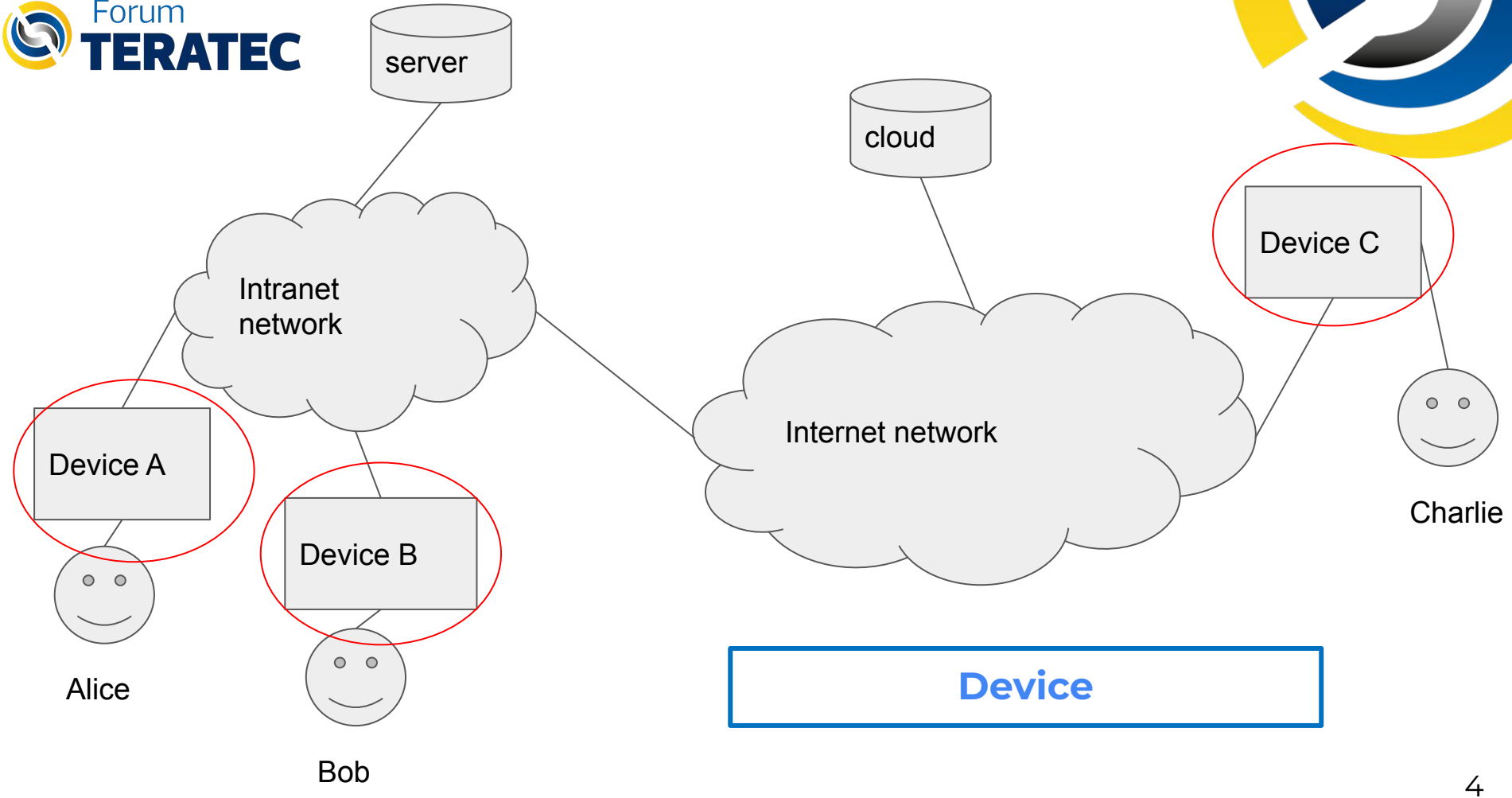
Forum TERATEC

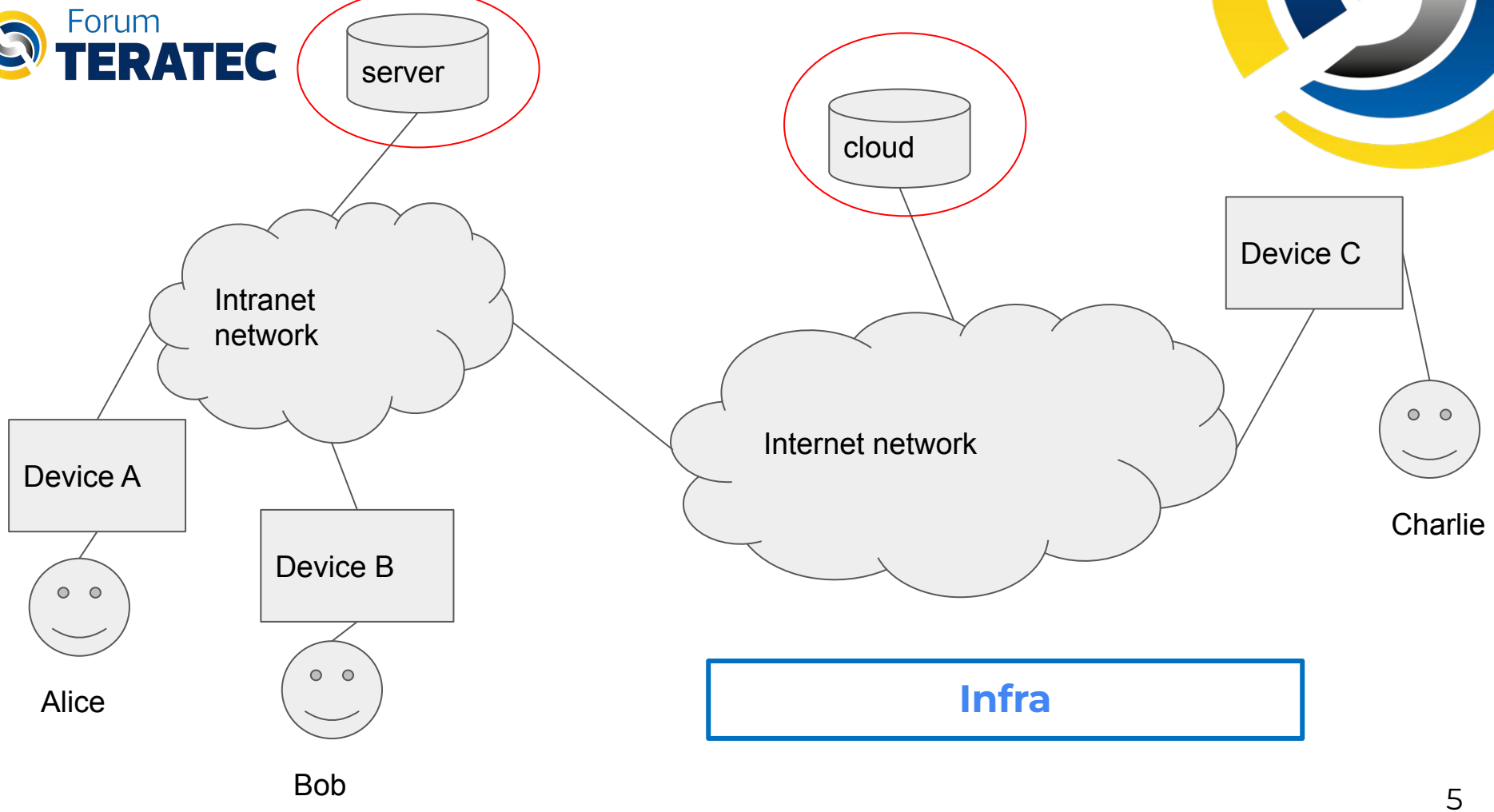
Zero trust Data Centric Security : What is this ?

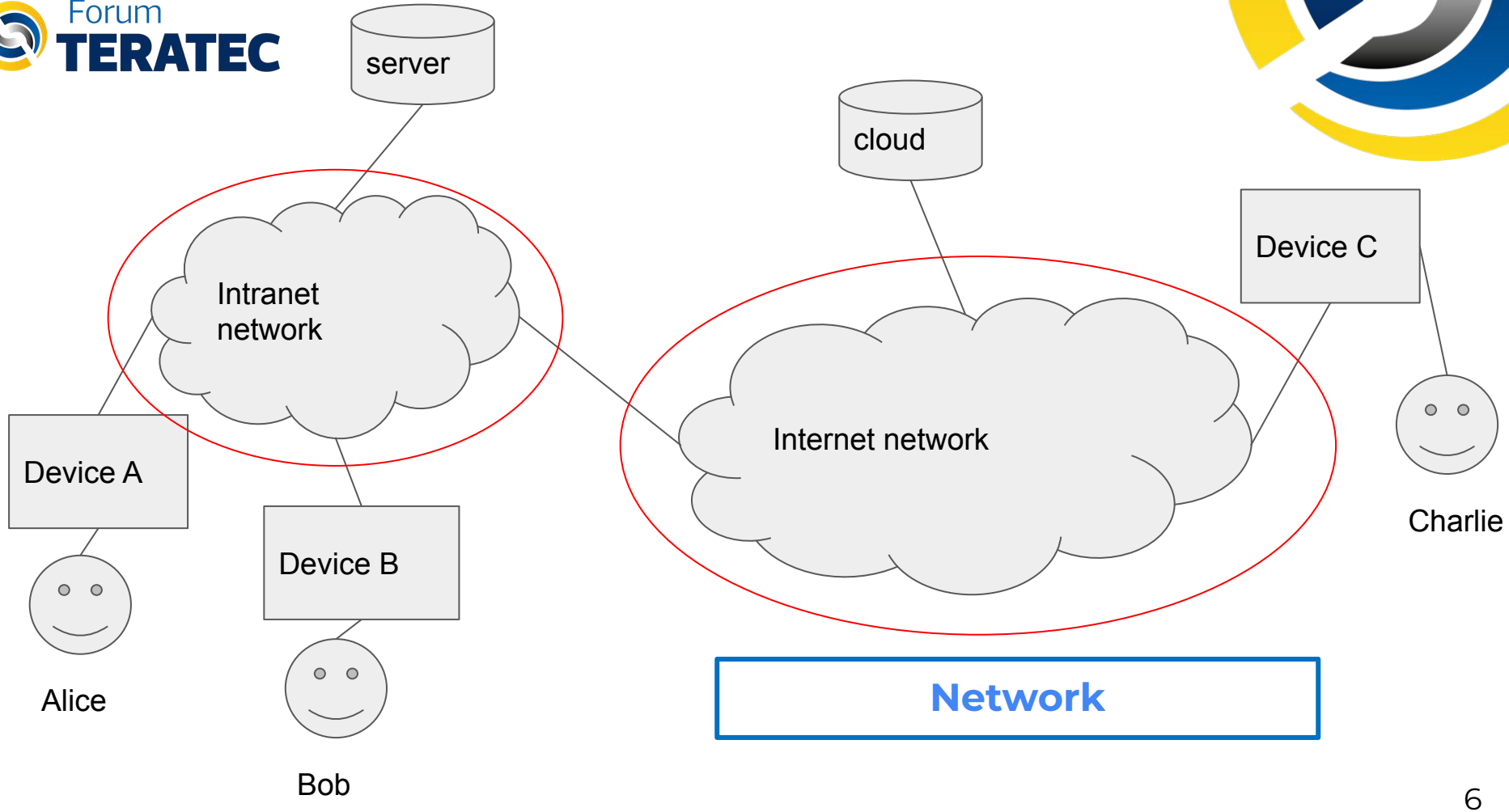
- Security is no longer an option:
  - Obsolescence of the perimeter security model
  - NIS2 directive applicable end 2024
  - Zero Trust DoD strategy / US NIST standards
  - Nato DCS standardization
- Requirements
  - security as close as possible to the user
  - confidentiality, integrity, non-repudiation, authenticity, anonymization, traceability, historization, revocation
- Means
  - Zero trust ⇒ systematic verification
  - Zero knowledge ⇒ crypto implementation of "right to know".
  - Data Centric Security ⇒ crypto-control of data & classification metadata

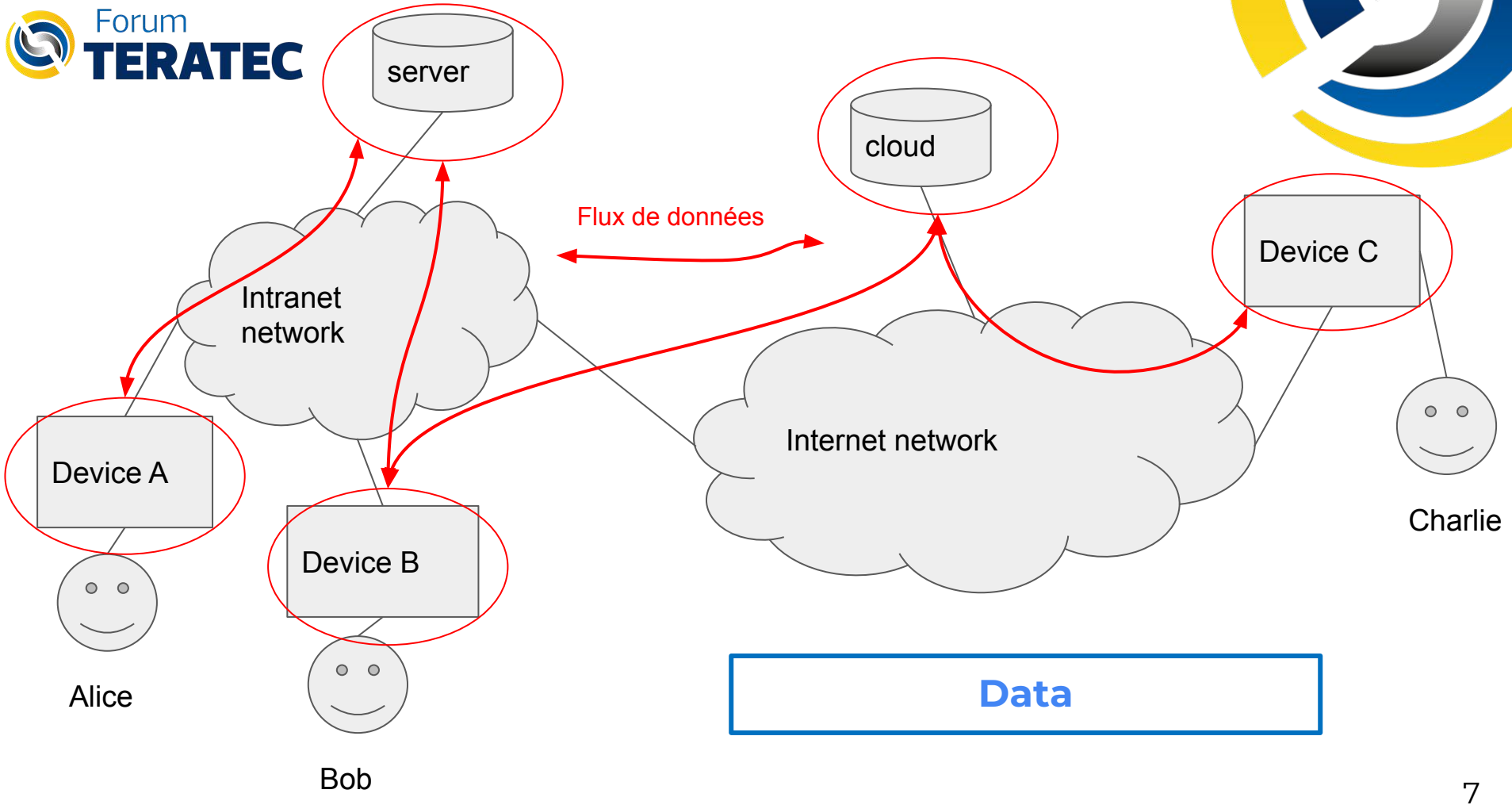

server
cloud
Intranet network
Device C
Internet network
Device A
Device B
Charlie
Alice
Bob
System


server
cloud
Intranet network
Internet network
Device A
Device B
Device C
Alice
Bob
Charlie
User


server
cloud
Intranet network
Internet network
Device A
Device B
Device C
Alice
Bob
Charlie
Device

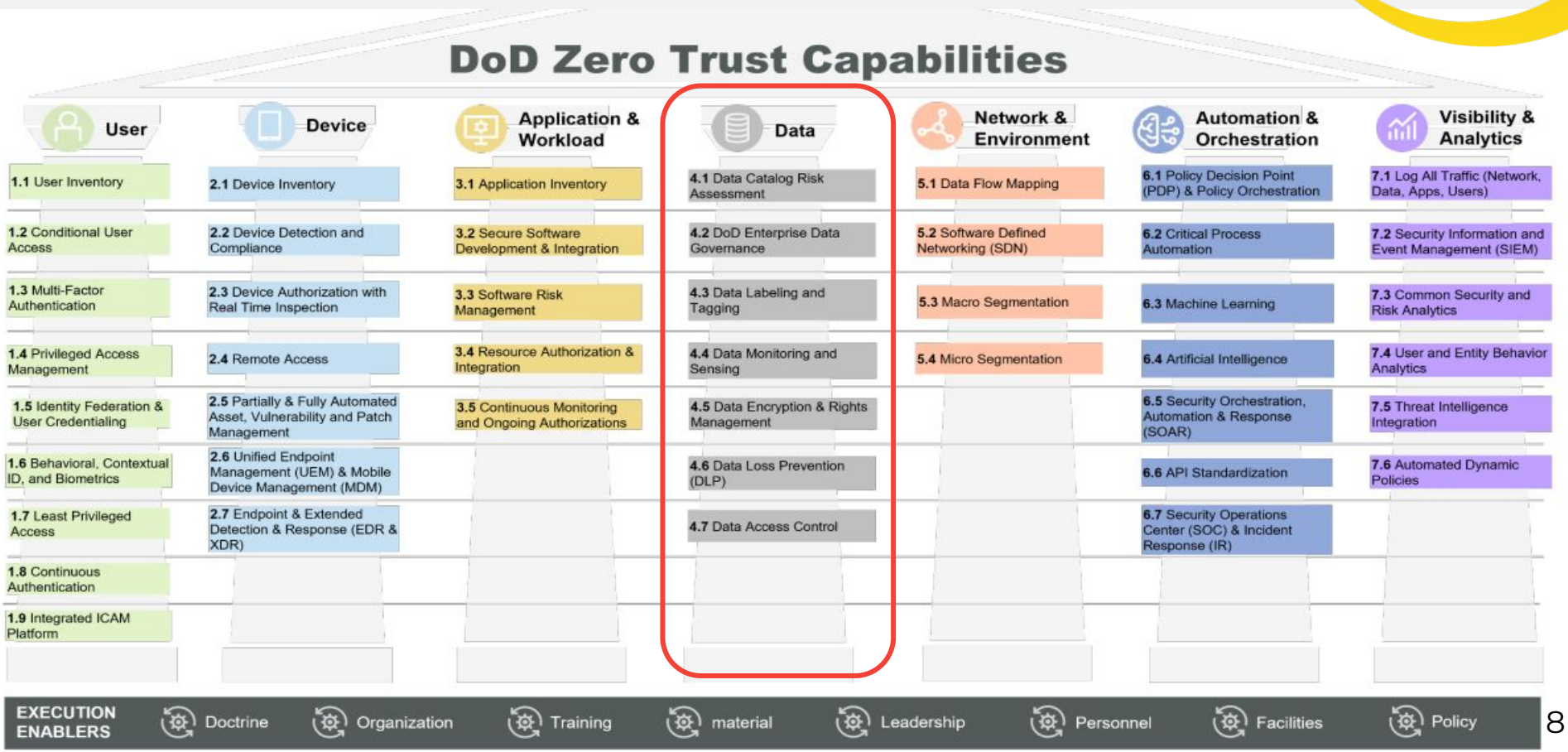

server
cloud
Intranet network
Internet network
Device A
Device B
Device C
Alice
Bob
Charlie
Infra


server
cloud
Intranet network
Internet network
Device C
Device A
Device B
Charlie
Alice
Bob
Network

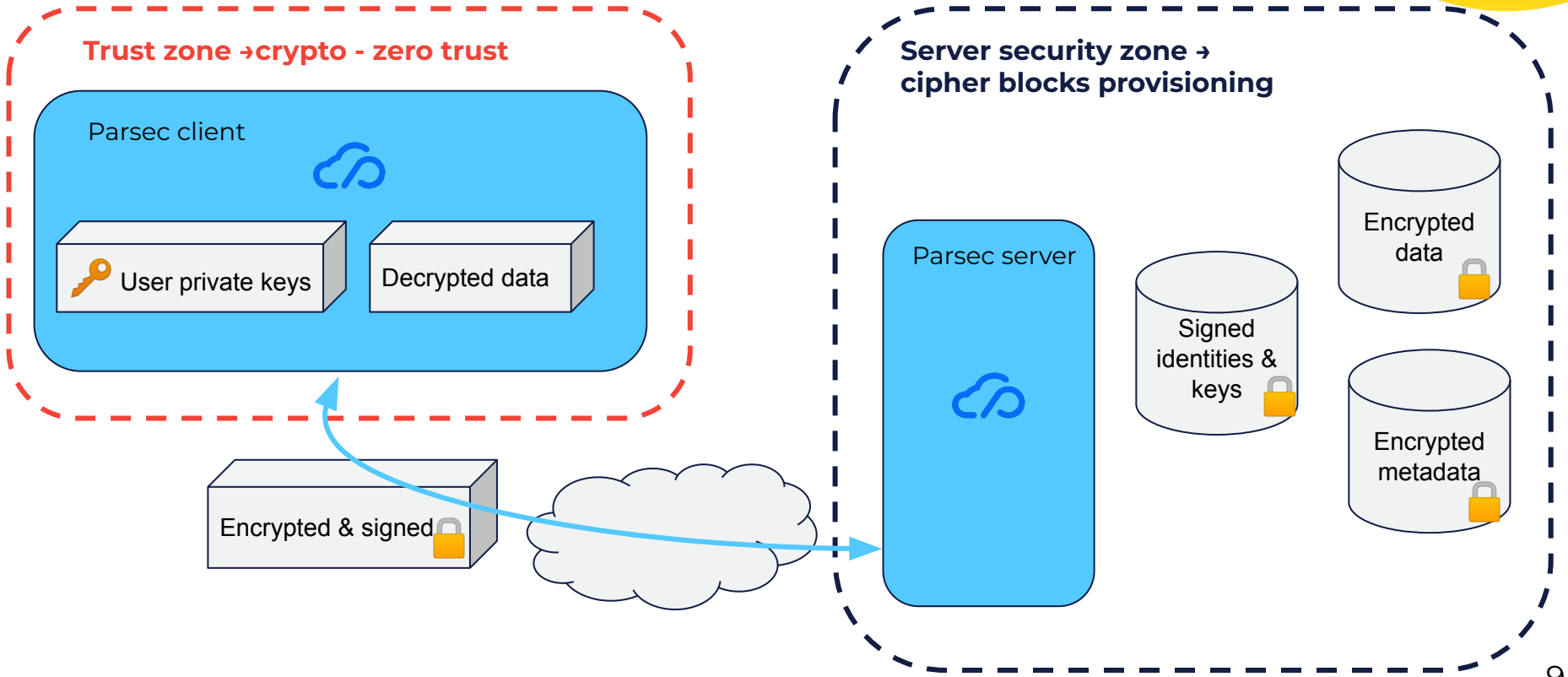

server
cloud
Flux de données
Intranet network
Device C
Internet network
Device A
Device B
Charlie
Alice
Bob
Data

Forum TERATEC

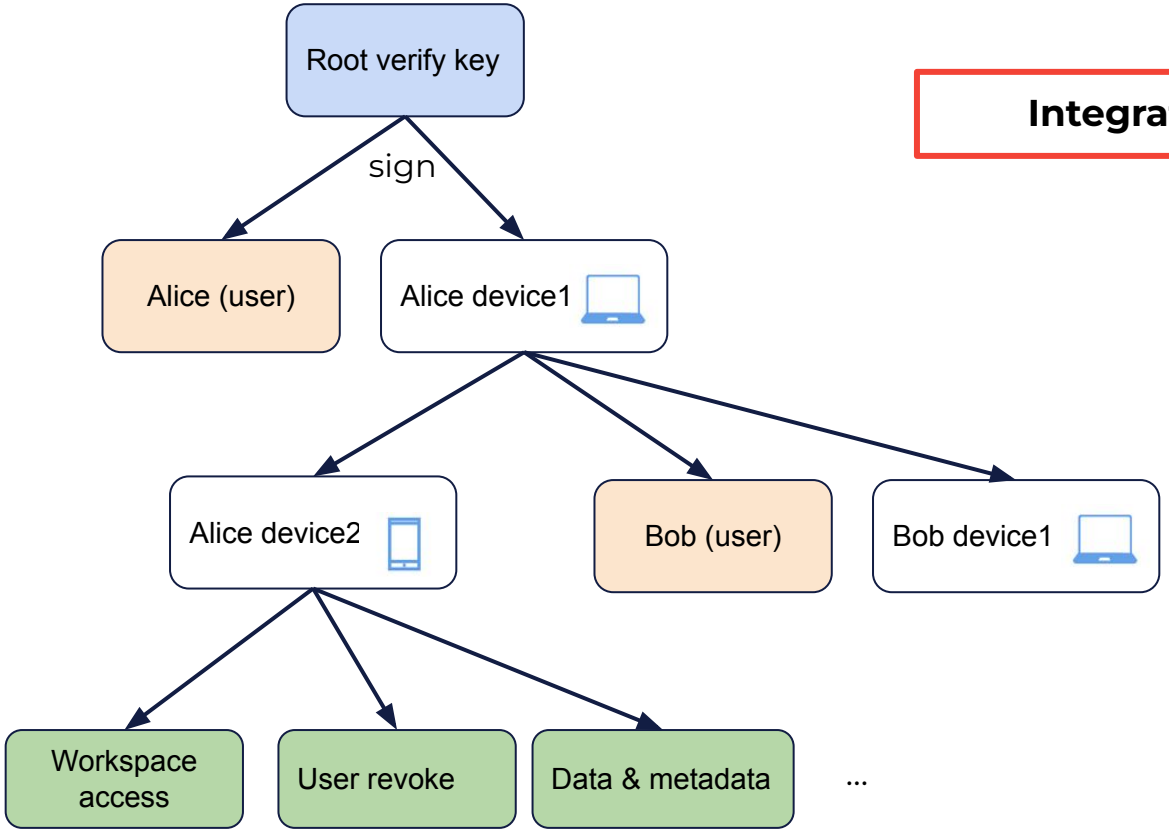
US DoD standard Zero Trust



Parsec: Systematic user-centric encryption

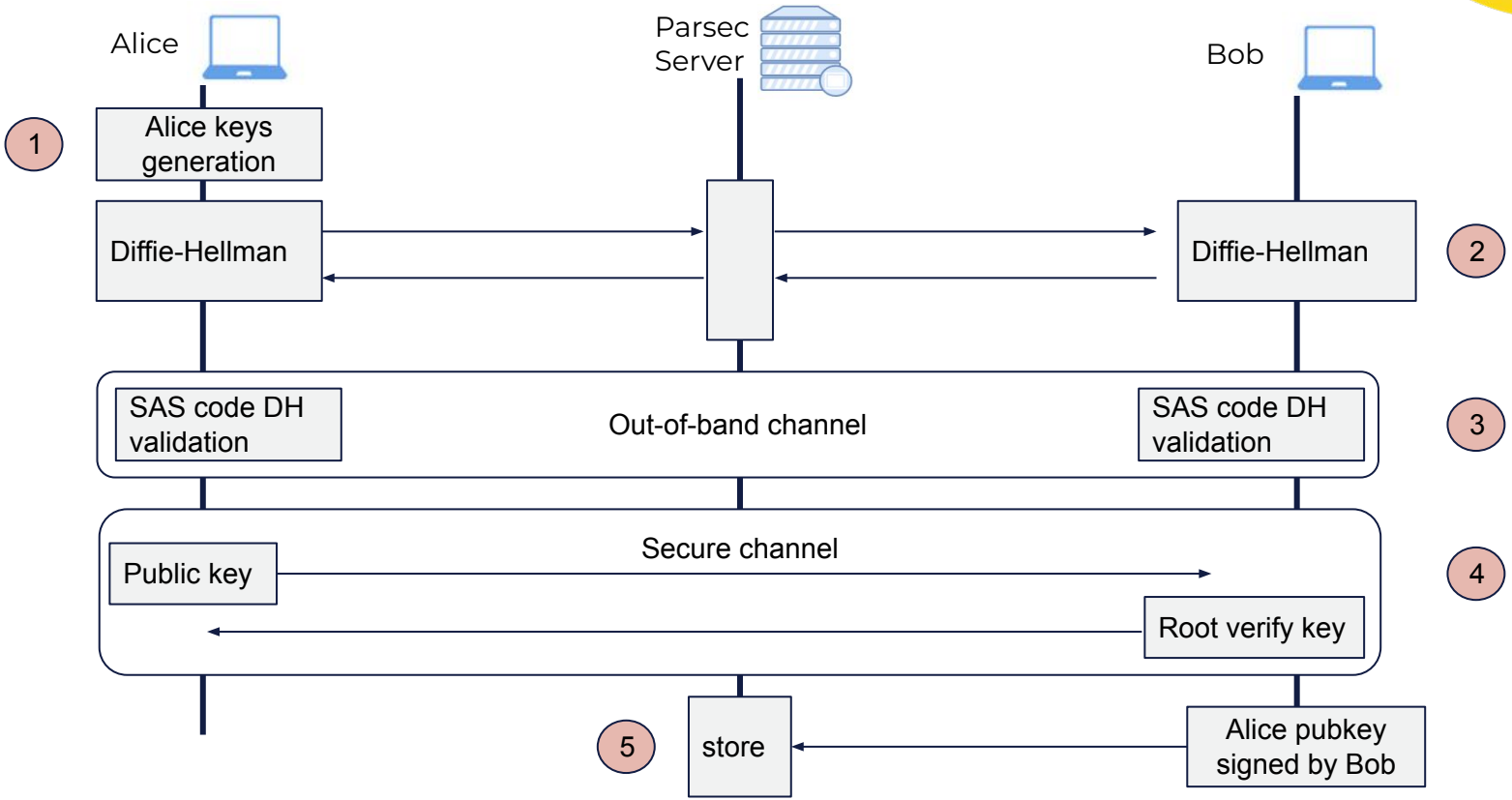


1) Circle of Trust management

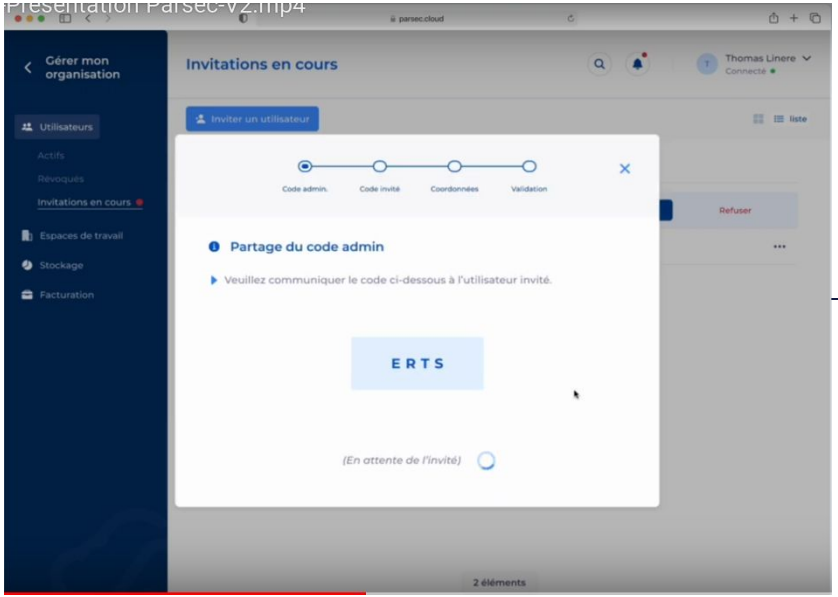


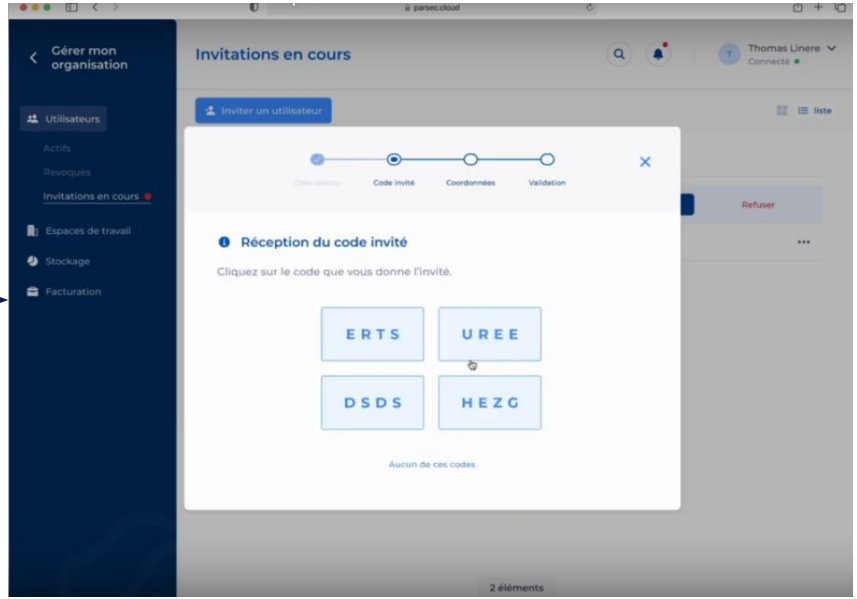
Integrate a dedicated PKI.

2) Simplified user enrollment & dedicated PKI



2) Simplified user enrollment & dedicated PKI

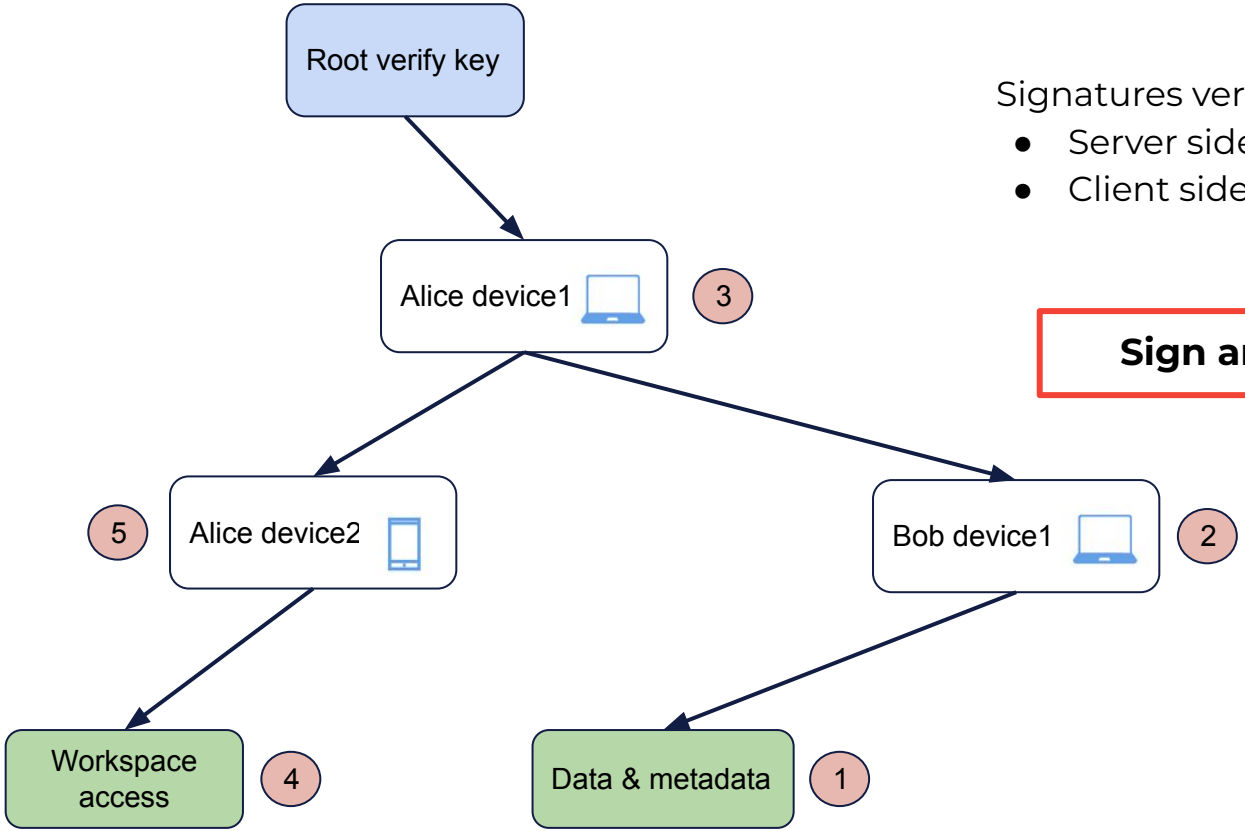




Only humans creates trust.

https://drive.google.com/file/d/1_abgGYTzRJwuTshZDQ0JMGnwb5_Opgyt/edit

3) Zero-trust data access

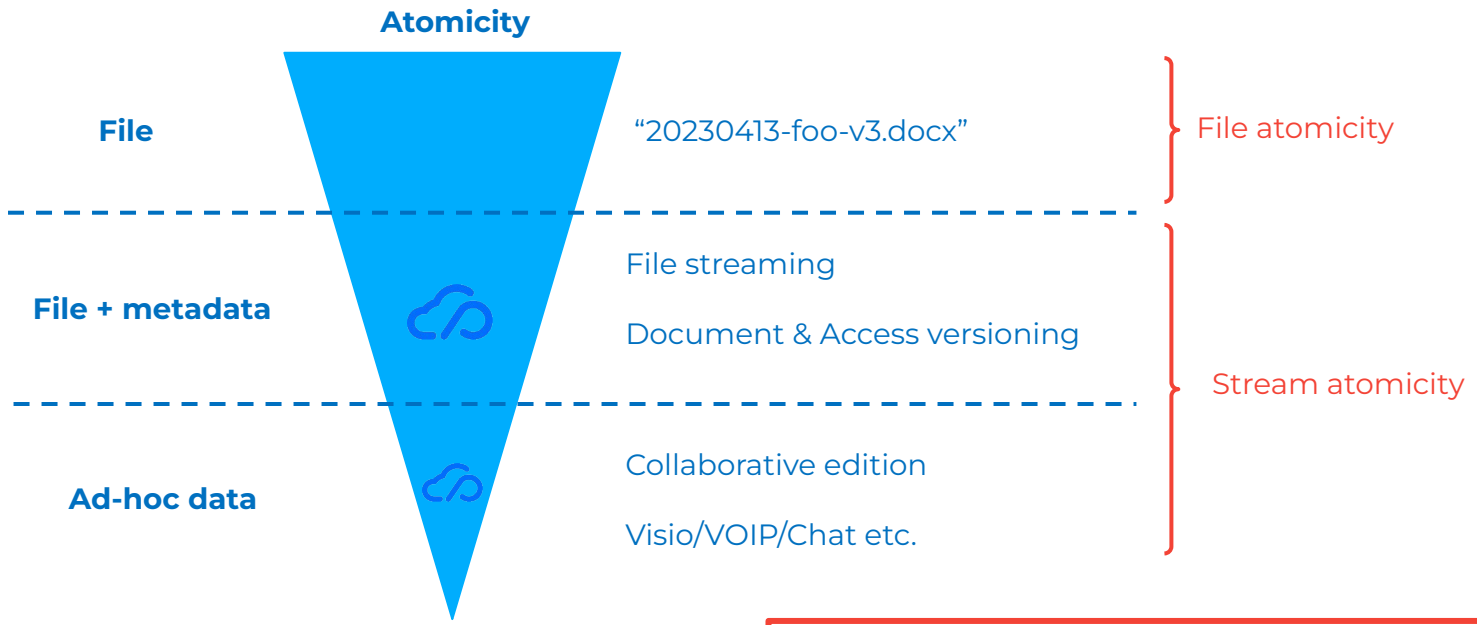


Signatures verified:

- Server side(access control)
- Client side (end-to-end trust)

Sign and encrypt everything.

4) File versus stream atomicity



Process flows, not documents.

5) *Challenge 1 :* Backward compatibility

In a classic web application:

- update functionalities → evolve the data model
- very simple on a centralized database

In an approach that no longer trusts the central server:

- All data is signed and encrypted, and the central server sees nothing (we don't trust it).
- How do you add a new concept (e.g., adding backward-compatible data classification)?
  - at server level
  - at client level: who signs?

<u>Challenge n°1</u> : the end-to-end encryption issue exacerbates the resolution of data backward compatibility.

6) *Challenge 2:* Strong applicative security integration

In a classic web application:

- security layer: TLS
- application layer: REST API

In a Data Centric Security application :

- REST API with signed & encrypted data
- Use case: Synchronization of (large!) Parsec files
  - divide file data into blocks
  - upload encrypted blocks to Parsec server
  - create a file manifest to reconstruct a given version of the file
  - file manifest signed and encrypted (with workspace key), uploaded to Parsec server
  - ⇒ problematic: user access revocation (file manifest re-encryption without signature change)
  - ⇒ upload order between blocks and file manifest

<u>Challenge n°2</u> : strong integration between security and applications.

7) Eventual consistancy

In a classic web application:

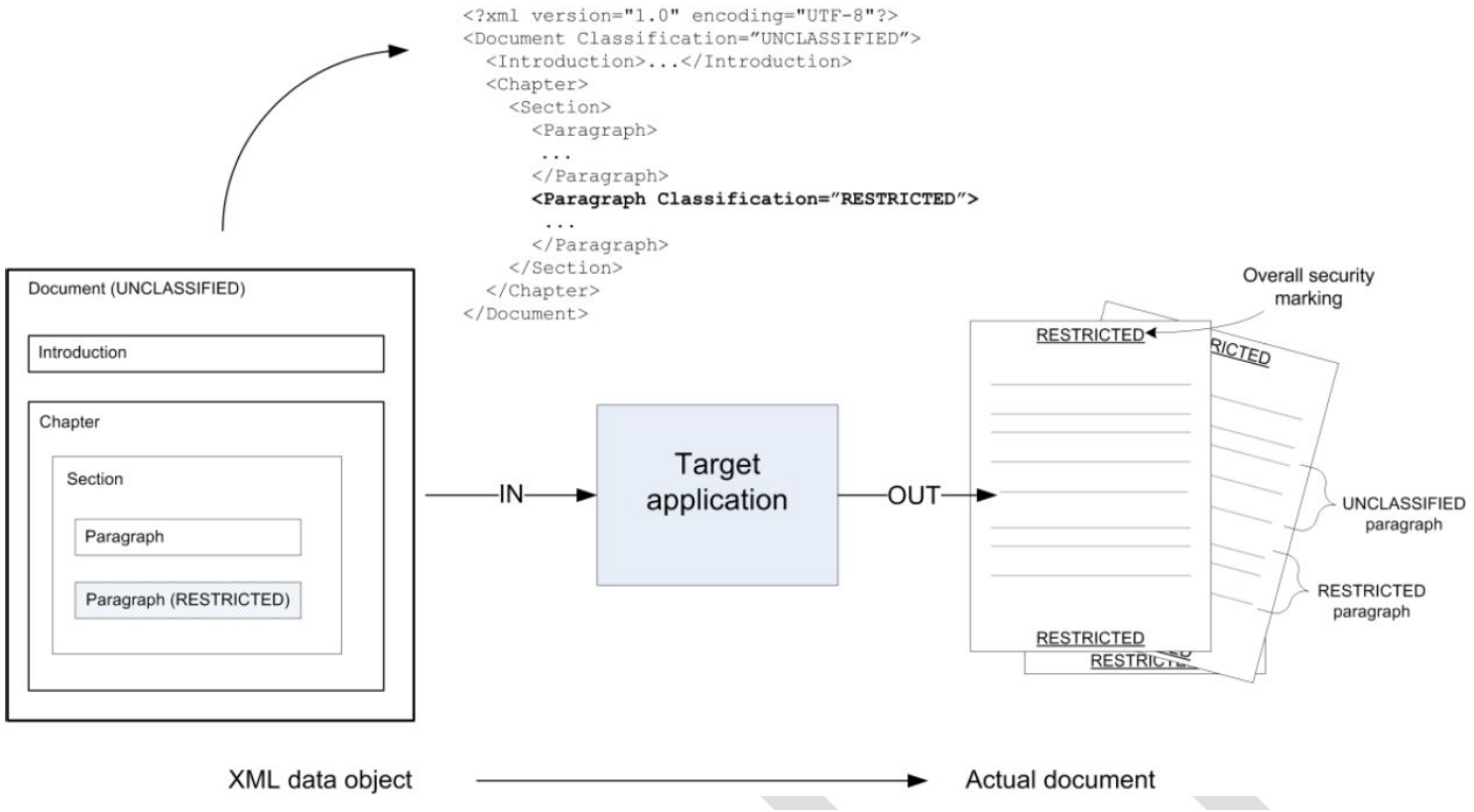
- he client is seen as the graphical interface
- a request to the server is enough to modify system state

In a PARSEC logic, client and server states must be reconciled

- Issue 1) = withstand network failure
  - survive loss of connection
- Issue 2) = tolerate network latency and manage time scales
  - constant de-correlation between server state and client state
  - local modification must remain instantaneous as seen by the user
- Consequence: asynchronous operating mode:
  - operations are always performed locally
  - network outages have no consequences
  - client and server systems accept the decollation and agree when the network connection is re-established

Challenge n°3 : correlate server and client status despite latency and network interruptions.

DCS NATO vision of data labelling



DCS NATO maturity levels

- DCS#1 : Labeling
- DCS#2 : Trusted binding (label signature))
- **DCS#3 : Signed and ciphered data & metada ta**

NATO inter-nation DCS architecture