intertrust

# Global Distributed Trust Management

## Using distributed ledgers

**Dave Maher**
EVP, CTO Intertrust Technologies

September 2018

# Thesis

- In order to provide security and trust for the burgeoning IoT & IoE, the current digital certificate based trust infrastructure needs to be replaced

- We should replace it with a more responsive, highly scalable, non-hierarchical, distributed trust management fabric

- This fabric can consist of heterogeneous deployments of many independently operated, lightweight, distributed ledgers. Each would be dedicated to support specific kinds of trust assertions from trusted knowledge sources, and *capable of rapidly responding to market needs*

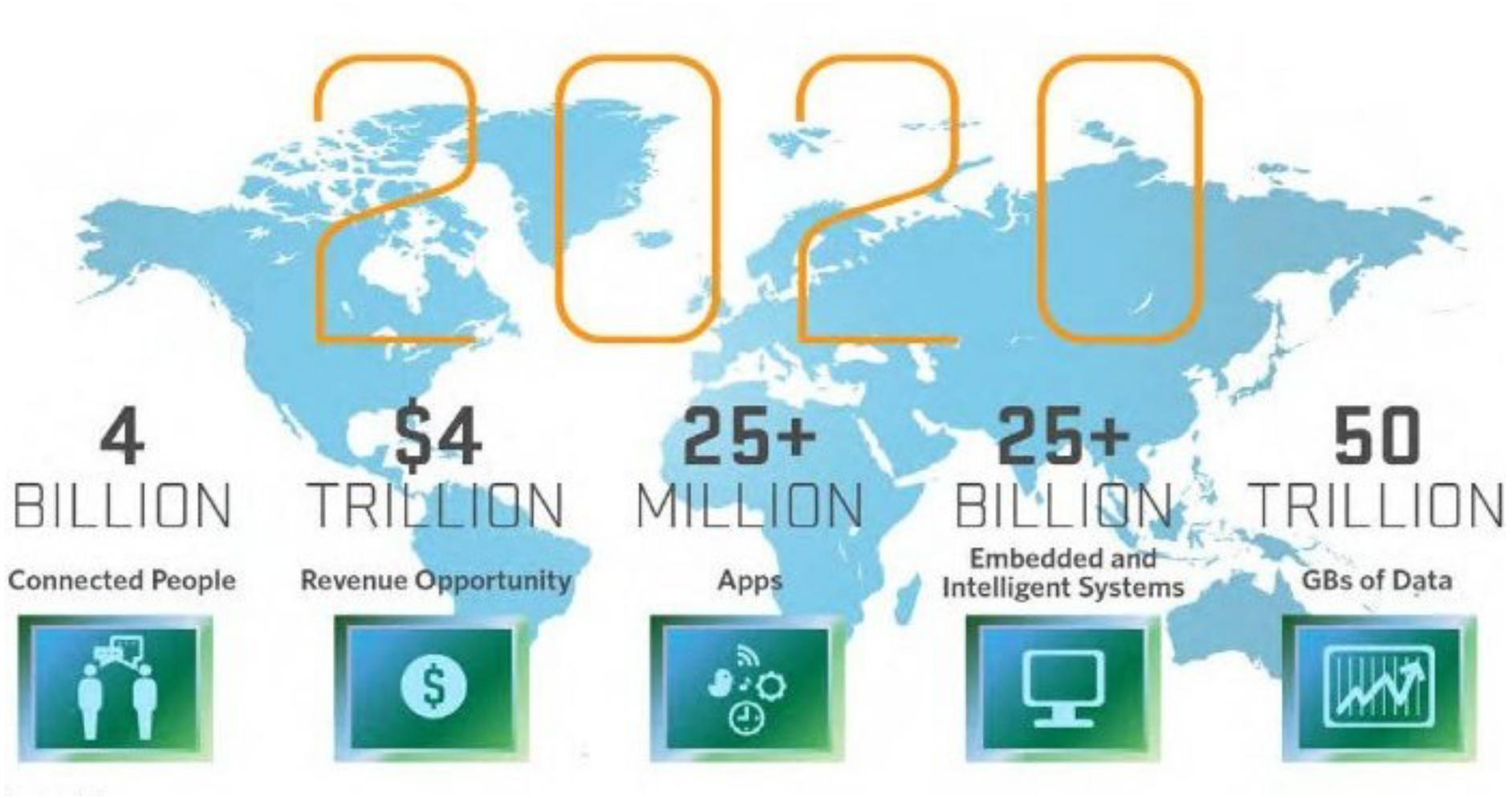
# What is so profoundly different about IoT from a Trust POV?

- Scale and Ubiquity
- Hyperconnectivity
- Implications of merging the Cyber and Physical worlds
- New interaction modes

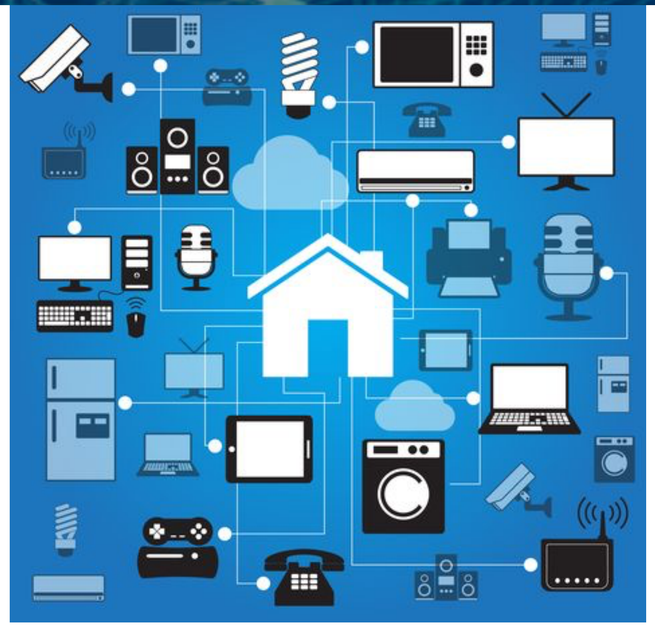
*Old approaches and strategies won't work*

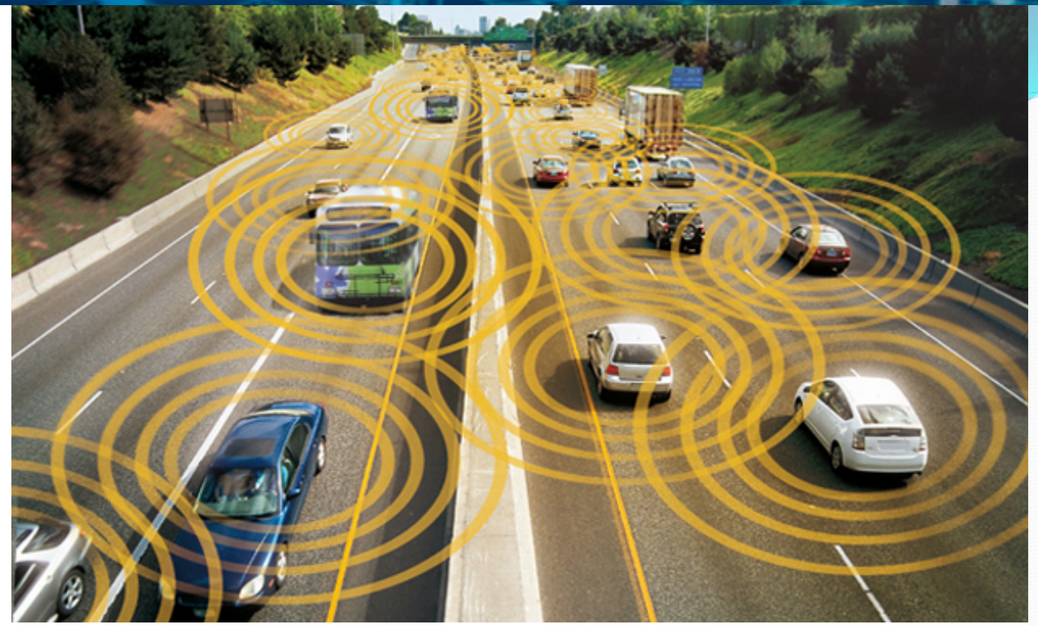
*Real innovation is required*

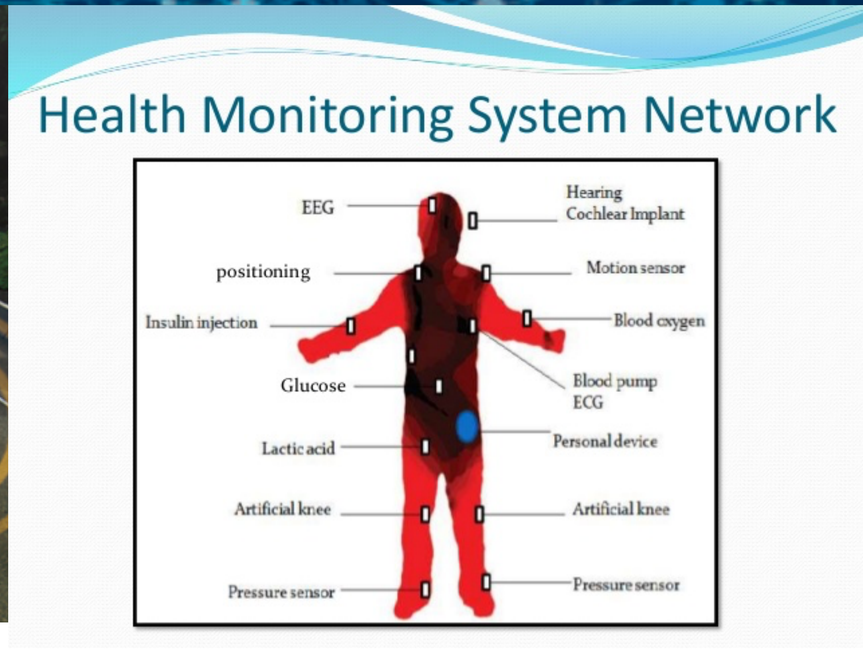
# Scale and Ubiquity



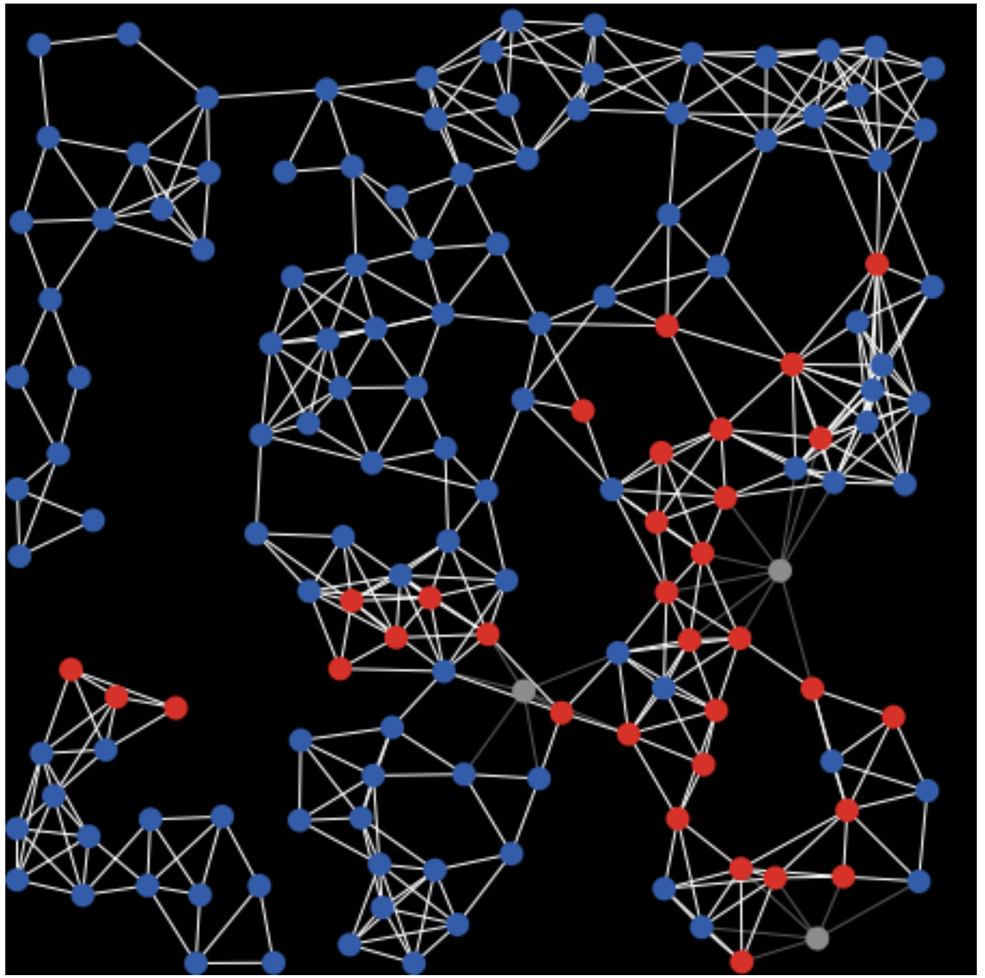
# Hyperconnectivity and dynamic and ephemeral networks







# When everything is connected; network security won't work



# Merger of Cyber and Physical worlds brings huuuuge risks

# IoT brings new interaction models

- Everything can be controlled from everywhere through many paths
- Devices will have digital twins (virtual copies) located in the cloud or in gateways
- Virtual devices can be composed and people will interact with dynamically defined command interfaces with different effects at different times
  - Press a button and different things happen than yesterday
- As devices are added to a system, they will form interactions with one another, often automatically — Machine to machine interactions will increase dramatically

# Massive automation drives IoT

- Actions and decisions need to be made on the basis of valid and timely info
- Decision policies will incrementally improve and reference new and better information sources
  - This makes it difficult to anticipate needs as part of design-by-committee standards process
- More entities need to trust one another in different contexts according to compliance policies

# What kind of knowledge needs to be applied?

- Valid device identities will provide safety critical information necessary to make decisions or take action e.g. automotive V2V, V2X, industrial IoT, public safety IoT. These device identities will include:
  - Identity attribute bindings, such as public key, network address, title, authority, clearance, biometric hash
  - Device info
  - Licenses and certifications
  - Compliance affirmations
  - Data tags: ownership, provenance info (GPS, sensor ID, integrity hash, etc.), security level, privacy status

# Whats wrong with X.509 certs?

- Certs are hard to manage, compromise recovery is hard
  - When keys are compromised or suspected of compromise, everything they signed in the past becomes suspect (no PFI), so cert chains and intermediate keys are added, increasing complexity
  - Difficult to manage change when the underlying assertions that the cert attests to change; revocation has never worked well as studies show
  - Relatively inefficient compared to other technologies; can be hard to scale
  - Tend to rely on centralized authorities who may not have the best and most up-to-date knowledge needed to make the best assertions about a topic
- Newer technologies are available that allow the binding of keys to identities and attributes
- Some of those technologies are gaining widespread support for many security related use cases giving us a good body of knowledge, talent, and tools

# Another promising tool: Assertion-oriented Blockchains

- Universal Trusted Oracle
  - Is *this* statement true? What do other people or entities I trust say about something?
- Provenance and authenticity for sensor data recordings
  - Telemetry from devices
  - Event streams
  - Video recordings and photographs
  - Has that photograph been modified?
- Immutable information tagging stamps of approval for Privacy, Safety, Security, and Trust
  - Tag data files with attributes of ownership, confidentiality, provenance,…
- Can authenticate security associations, replacing certificates

# An efficient infrastructure for recording assertions

- Assertion A about a subject made by entity E is hashed and signed by E with Secret Key S and entered into a distributed ledger
- When I want to rely on assertion A, I use a "root policy" or set of fundamental beliefs I maintain, and a set of required observations I can verify from the ledger in the form of previously entered assertions
- Such assertions may include:
  - P is E's public key according to Witnesses W1, W2, W3,…
  - E is trusted to make assertions about the subject according to Authorities A1, A2, A3,…, as recorded in this or other ledgers
  - Assertions made by others about the Witnesses and Authorities
- Recursively, more assertions can be referenced as needed to verify the previous assertions

# TIDALs: Trusted Immutable Distributed Assertion Ledgers

- Trusted because you rely on the Ledger (or its derivatives) for important things
  - Blockchain transactions are said to be trustless; we don't make this claim
- Immutable (helpful for audits) — what happened cannot unhappen; helps provide Perfect Forward Integrity for bindings and data
- Distributed: Distributed trust and distributed, synchronized information locality
- Assertion: All kinds of assertions, but typically simple bindings of information
- Ledger: Permanent recording; accessible to public or authorized subset

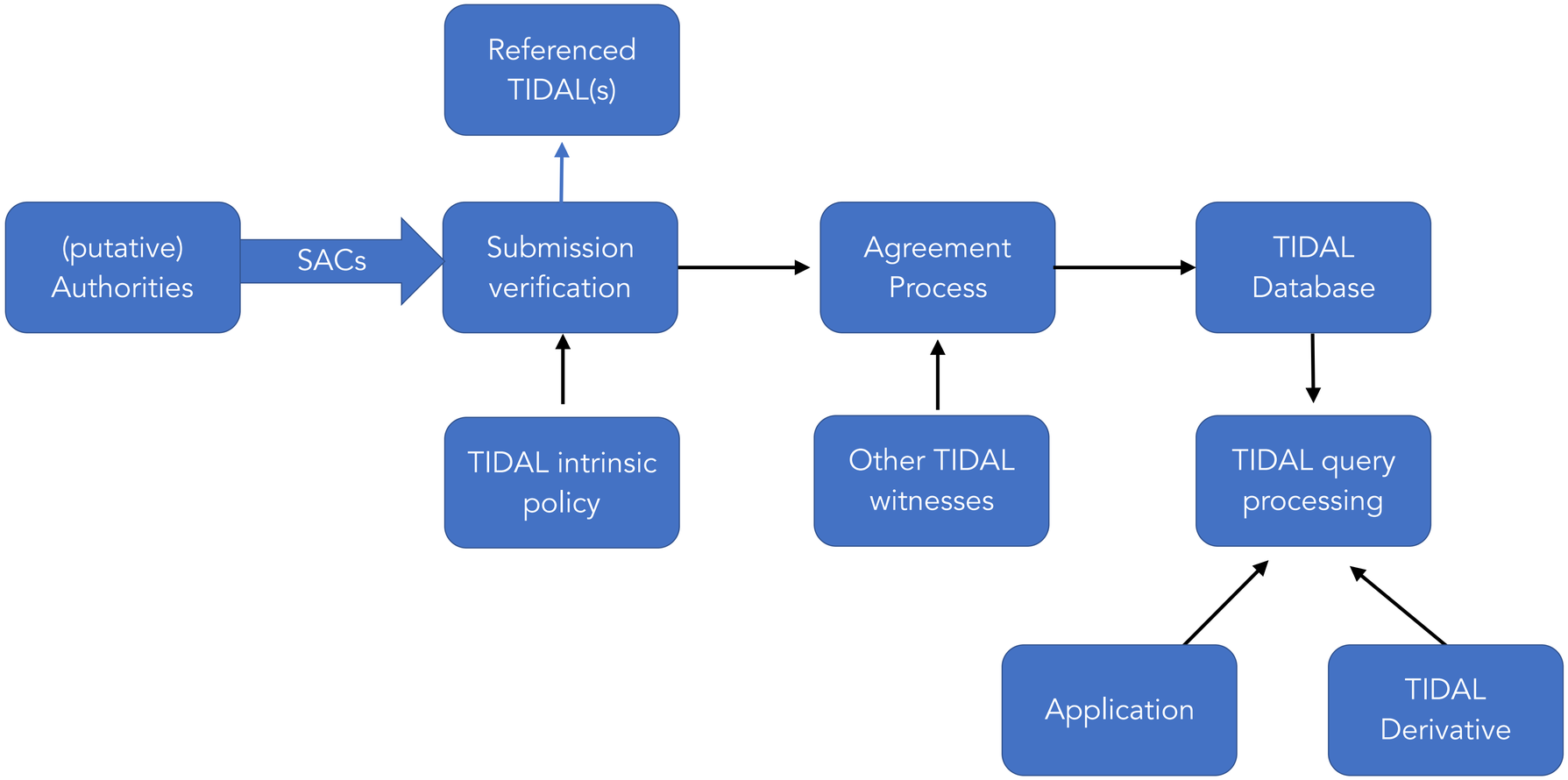
# TIDAL derivatives

- Contain an index or a hash table for easy reference to a subset of records
- Can be distributed and synchronized, cross-checking one another
- Can be constructed and maintained by entities focusing on specific types of applications
- Typically only need to be an ordered Hash Table
- All the world's web server bindings fit into a 32 GB hash table
  - 1 Billion 256 bit hashes
  - 1 lookup (for a Freshkey protocol instance is, on average, 30 1 byte compares)

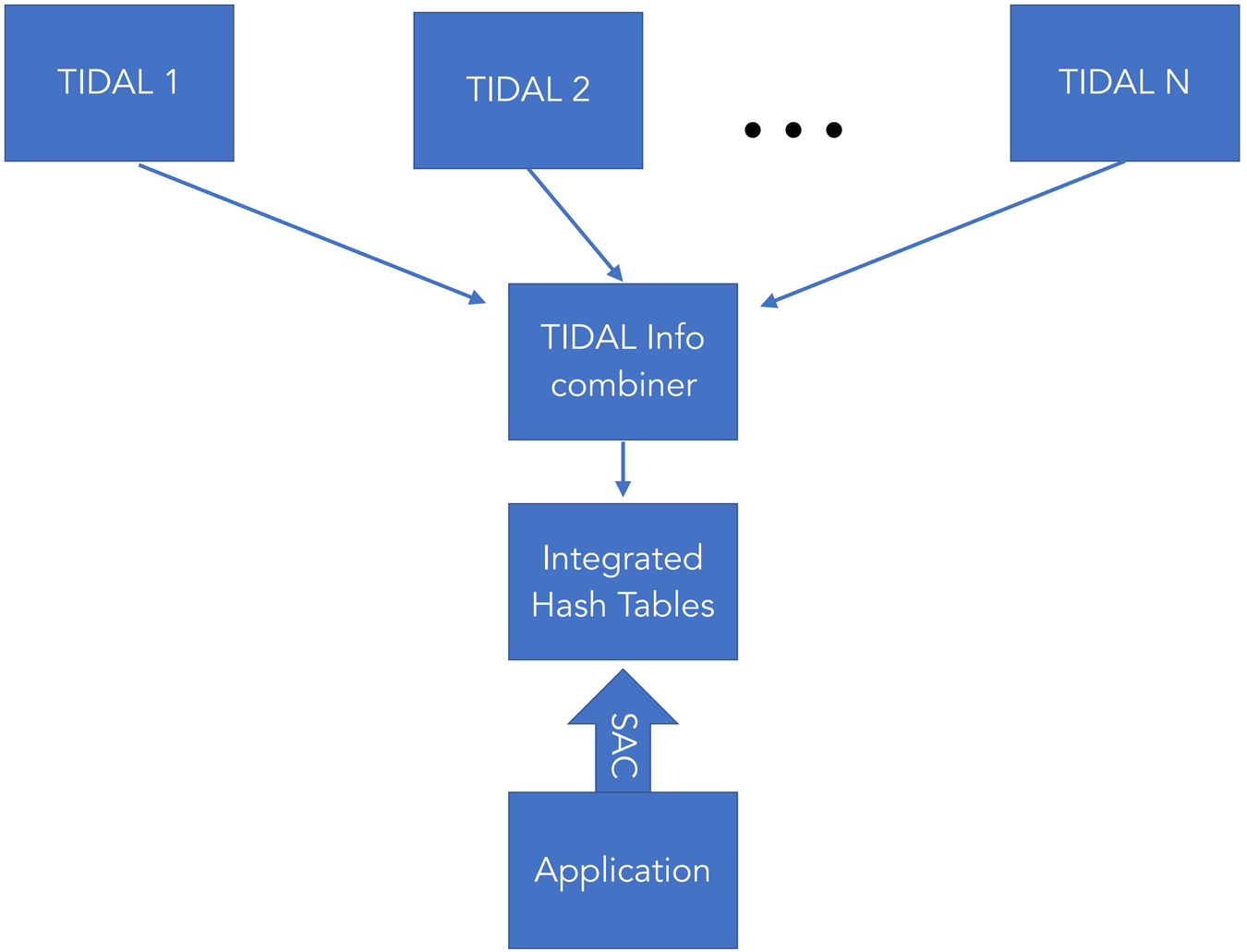
# Roles in TIDALs

- **Domains**: Labeled, registered areas of authority
- **Subjects**: Labeled, registered topics (areas of interest) within a domain; will include attribute names
- **TIDAL internal (intrinsic) policy**: rules defining who has the authority to make assertions about subjects in a domain
    - As opposed to **extrinsic policy** that can determine what actions the user of a TIDAL might allow
- **Attesters**: Principals that make assertions about a subject within the context of a domain
- **Validator/Verifiers**: Entities that verify assertions submitted by an attester
    - They verify the assertion was made by the attester and (using intrinsic Policy) that the attester has the authority to make the assertion about a subject
- **Node operators**: Entities that maintain a copy of the database containing the TIDAL and agree to faithfully follow protocols to update their copies (they may also be validators)

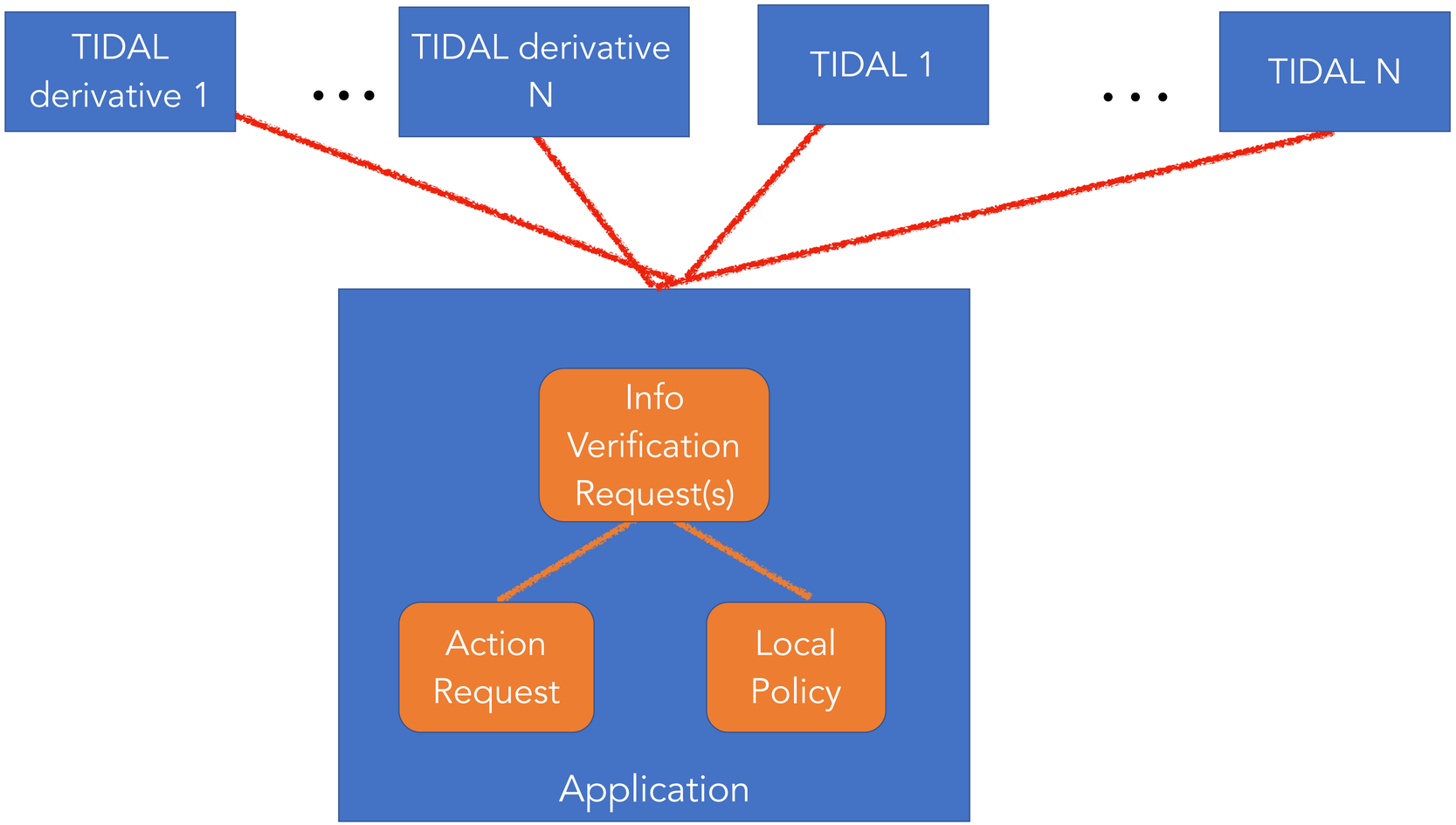
# Processing of trusted assertions in a TIDAL



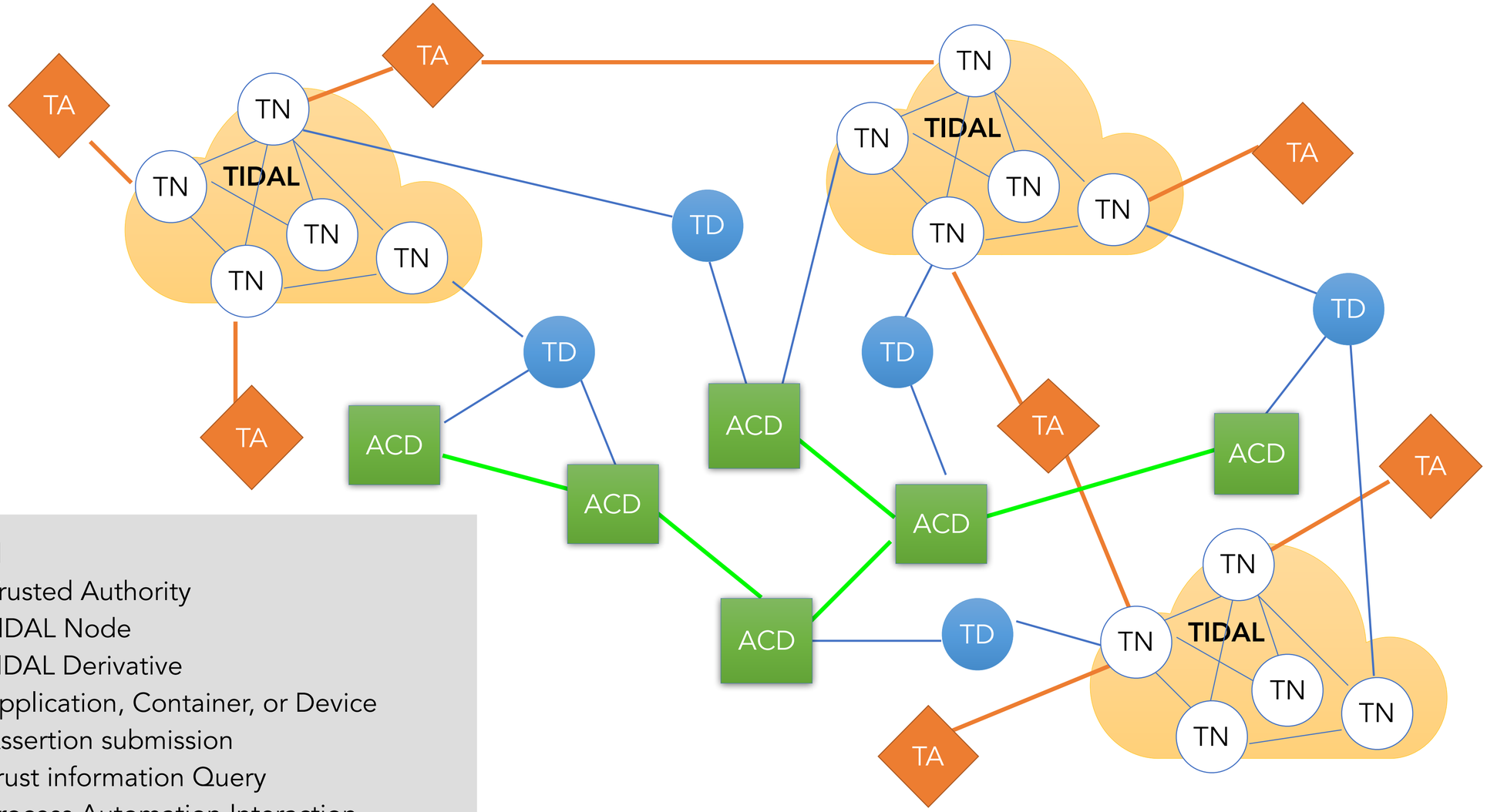
# Application referencing a multiple authorities through a TIDAL Derivative



# Application seeking authorization for action



# Global Trust Infrastructure with many specialized TIDALs



# Web trust example

TIDAL A records affirmations linking domain names and IPv6 addresses (DNS server). A TIDAL entry includes among other fields, a hash

H1= h(IPv6_address | domain_name)

TIDAL B records affirmations linking domain names and thumbprints of public keys for the servers using those domain names, computing

H2 = h(h(public key)| domain_name)

TIDAL derivative D monitors TIDALs A and B removes revoked entries and computes new

H3=h(H1|H2)

**A single hash table lookup provides fresh information on**

- **DNS security**
- **TLS security**

Is H3’ in the hash table of D?

Application visits a web server
policy: check IP address and TLS key
Computes H3’ of relevant info

Web server

# Can a digital photo be believed?

- Need more than just a digital signature on the photograph
- Infrastructure can be used to make assertions that strengthen trust in sensor information **and** in the chain of handling and control
- The following assertions (and more) can be made to check provenance
  - Authority1 endorses a public key used by a sensor module consisting of
    - Photographic Sensor
    - GPS sensor
    - Software postprocessor
  - Authority2 endorses a public key for a photo processing entity
  - Authority3 endorses (signs) photo post-processing software
- Forensic process can check each endorsement and credentials of each authority using blockchain, following an end-to-end chain of handling and control

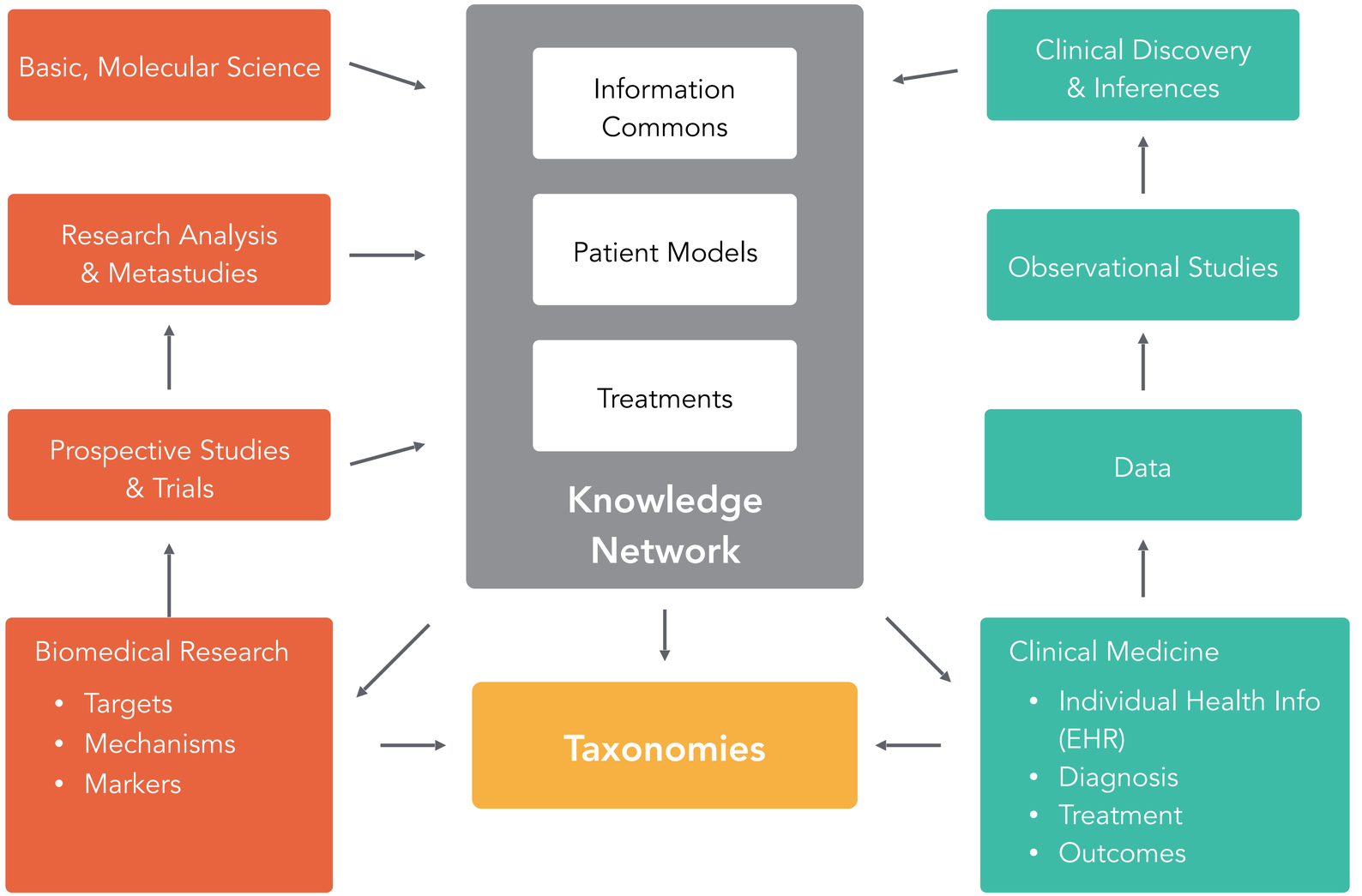
# Implementations – Driving for Simplicity

- Each TIDAL can typically be a very simple database with blockchain
  - Permissioned, with an intrinsic purpose-built policy for recording
  - Submitter's credentials will be checked by referencing other TIDALs where those credentials are registered
  - Number of nodes and Byzantine agreement protocol appropriate to the kind of info stored in that TIDAL and the threat model for that info
- Easy to implement with Hyperledger and other open source projects
- Derivatives are typically very compact and efficient hash tables but can include other information to aid in error processing and multiple derivative copies can cross-check one another
- Stateless APIs can be standardized
- Policy languages can be simple propositional logic on name value pairs or regular expressions

# Diverse, specialized TIDALs that may be used by healthcare apps and personalized medicine practices

- Device info
- Compliance info
- Personnel directories
- Hospital, corporate information
- Health records and personal data including DNA sequences, clinical records, phenotypical data, etc
- Accrediting information
- Regulatory information
- Infrastructure service providers
- Treatments studies and Research

# With Security, Policies and Permissions we can enrich the Medical Knowledge Network with Clinical Data




**Slow Discovery**

**Fast Discovery**

# Cryptographic IDs and ID attributes

- Every device, software container, application instance, person can have a cryptographic ID (S, P, T) where
  - S is a random secret key
  - P is the derived public key
  - T is hash of P
- TIDALs can associate name-based identifiers, permissions, titles, licenses, etc. submitted by proper authorities by binding those attributes to thumbprints T

# Identity TIDALs binding crypto IDs to People, URIs, Device IDs can enable scalable Peer to Peer Security

- Efficient HTTPS channel authentication with secure DNS
  - Long-lasting credentials
  - Built-in revocation and renewal
- Every device with a TIDAL registered cryptographic ID implicitly has multiple secure channels with every other device without pre-arrangement
- We can implement reference monitors with TIDAL authenticated security associations (shared keys for authenticating commands and requests and data sharing)
- Rich identifiers can enable IFF (Identify friend or foe) protocols and private subnets, incremental discovery, and provisioning bootstrap protocols

# Authenticated Symmetric Key Establishment

## Requestor message

1. Protocol ID (string, including a version identifier)
2. Requester's time stamp
3. Request type
4. Requester ID info
5. Requester Long term public key or public key hash
6. Requester Ephemeral public key
7. Protocol specific parameters for the specific protocol ID

## Responder message

1. Protocol ID (can respond with a different protocol ID if allowed by the Protocol identified by the requestor)
2. Responder's time stamp
3. Response Type
4. Responder ID info
5. Responder Long Term public key or public key hash
6. Responder ephemeral public key
7. Protocol Specific parameters for the specific protocol ID

Stateless, layer-less; works in simple applications
Embeddable (e.g. in a database query)

# Key establishment procedure with PFS

- Each side hashes the info bindings as required by the protocol
  - Typically ID info and public key
  - Other parameter info can include roles, licenses, and other assertions from different TIDALs
- After checking with a TIDAL, construct the key:

$$K = h(b^{fe}|b^{MN}|\text{Requestor's time stamp} \mid \text{responder's time stamp})$$

or for anonymous client:

$$K = h(b^{fe}|b^{Ne}|\text{Requestor's time stamp} \mid \text{responder's time stamp})$$

Cache the key, cache the authenticity info

M is requestors private key
N is responder's private key
f and e are ephemeral keys
b is a DH group generator

Note: This works with ECC, R-LWE

# Compare

- Key establishment using certs:
  - Check the revocation list or make an OCSP call; Verify multiple certs in the cert chain for each binding (ID, role, etc.)
- Using TIDALs;
  - Search a hash table from a derivative; revocation checking is implicit
  - Hash table search is very efficient and generally Quantum Safe
- Compare Freshkey protocol vs X.509; IEEE P1363
  - No negotiation, but reliance on protocol registry
- Freshkey is embeddable, and queries can be answered according to policy by looking up requestor's credentials in TIDALs or TIDAL derivatives

# Conclusion

- Simple, economical TIDALs, based on blockchain and other distributed ledgers can efficiently replace digital certificates in this connected world
    - Security and Policy can be adapted for specialized knowledge
- Market-driven establishment of TIDALs and TIDAL derivatives with open APIs can allow information from hundreds of billions of authentic, knowledgeable sources and trillions of sensors to reach billions of application instances for reliable decision making and policy-driven actions
    - TIDAL recording focuses on collecting specific types of knowledge
    - TIDAL derivatives focus on organizing info from many TIDALs to efficiently serve applications
- Simpler, embeddable, stateless protocols can be used for simpler devices for application layer security and integrity, independent of network security


intertrust
BUILDING TRUST FOR THE CONNECTED WORLD