PHOENIX, AZ | MARCH 3-6, 2020

BLOCKCHAINS ARE NO LONGER MONOLITHIC PROGRAMS

They will become composable applications that meet a specific need

James Barry – CTO & Co-Founder
Bill Katsak – Chief Architect

What we will cover in 30 minutes

- Blockchain programs - Monolith program or a new way of programming?
- Can your blockchain adapt by workload?
- What would a Hyperledger stack look like today and tomorrow?
- How is Taekion using different projects to build out its application?
- What is missing from Hyperledger that can help build a custom fitted blockchain?
- Should Hyperledger be the Apache.org for Blockchains or Fabric with some add-ons?
- The future is bright for assembling an interchangeable stack of components

Taekion

https://www.theverge.com/2018/3/7/17091766/blockchain-bitcoin-ethereum-cryptocurrency-meaning

Blockchain as a Design Pattern

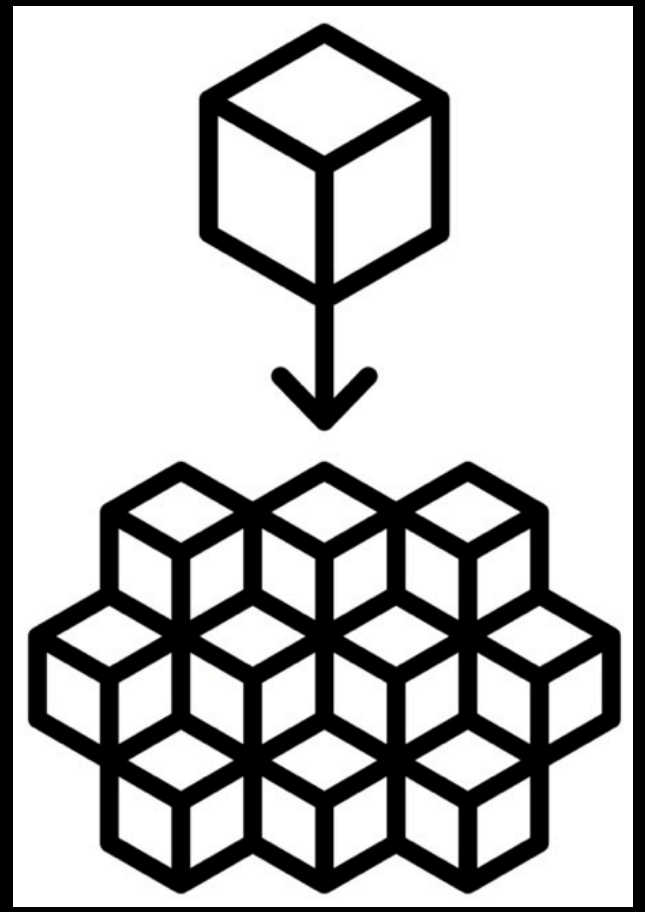
https://www.linkedin.com/pulse/blockchain-design-pattern-raghu-bala/

Taekion

WHAT DOES THE FUTURE OF BLOCKCHAIN HOLD?

Blockchain Monolith

Blown Up



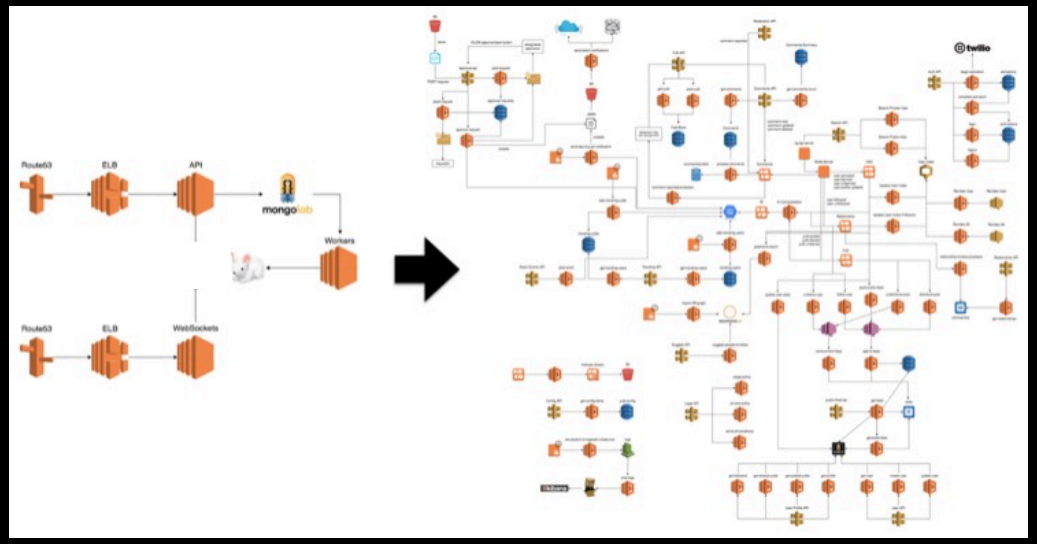
Into parts that fit your workload

Taekion

Standard Monolithic Software

OR

In the pattern of software decomposed by *Microservices like components*



Tomorrow blockchains will be assembled from best of breed parts that solve real world problems, or revolutionize industries

Think Facebook, Amazon, Google, Twitter, Alibaba

NOT

Netscape, AOL, Prodigy, MSN, Compuserve, eWorld

Taekion

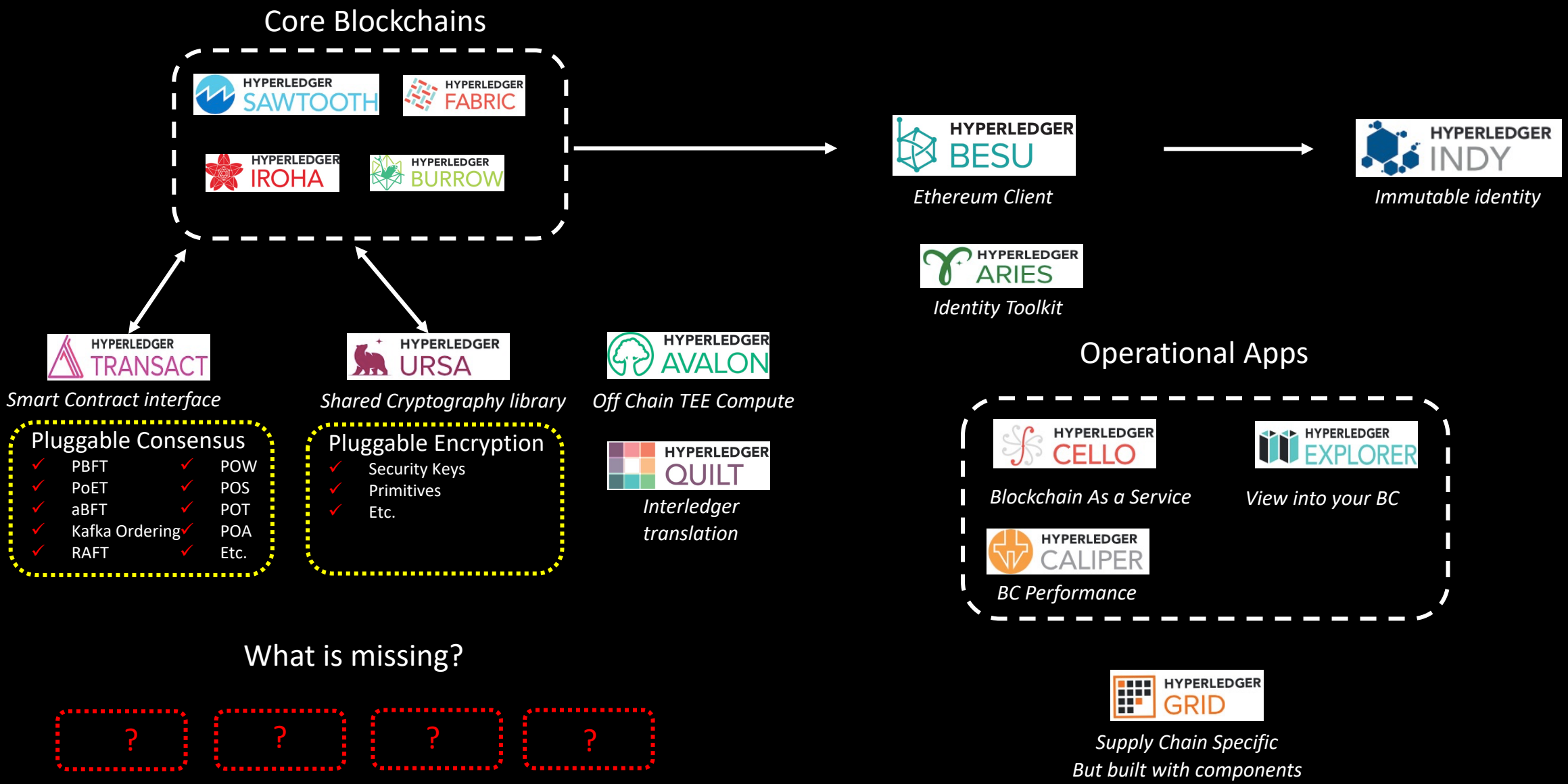
CONFIGURE BY WORKLOAD

Is one-size-fits-all good for Blockchains ?

	Private Network	Private Network	Public Network
High TPS	**Mass Market Product** *Stringent* *Consensus* *i.e. confirming no cheating for gaming*	**Private Network between subsidiaries** *Stringent* *Consensus* *i.e. Accounting settlement*	**Public Networks serving a worldwide consumer audience** *Irrevocable* *Transaction* *i.e. Crypto currency coins*
Low TPS	**Private Company** *Stringent* *Consensus* *i.e. Secure Files*	**Private Networks between companies** *Stringent* *Consensus* *i.e. Supply Chains*	**Public Network serving a small group** *Irrevocable* *Consensus* *i.e. Real Estate Titles*

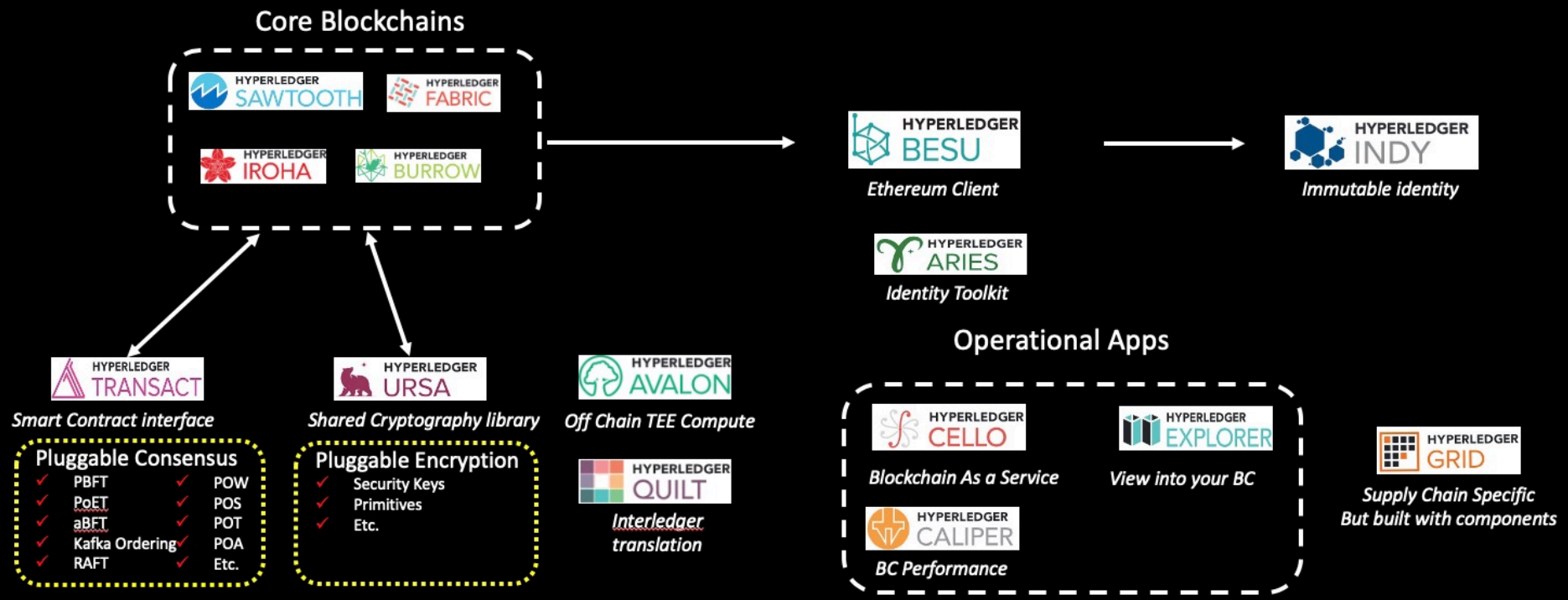
Taekion

SO WHAT ABOUT A HYPERLEDGER STACK?



Taekion

NEW COMPONENTS NEEDED FOR A COMPOSABLE CHAIN?



Block Storage

Network and connection management Layer

Key Storage

Operational Dashboard
Part Cello + Caliper for the business operator

Membership services

Block Archiving

Data Exporting

Data Importing
"Oracles connection"

Governance

Façade for business Logic
More than REST API

Permissioning/ Authentication

Token Management

Universal Wallet

Taekion

HOW TAEKION USES SAWTOOTH AND THE AVAILABLE LIBRARIES?

- **Sawtooth:**
  - Base of the Taekion Core platform.
  - Each core app has its own Transaction Processor (transaction family).
  - Transaction *batches* allow state to be manipulated across families atomically.
  - Signing and permission facility is very powerful: per-entity access control.
  - Pluggable consensus: different consensus for different use-cases.

Taekion

TAEKION USE OF SAWTOOTH AND THE AVAILABLE LIBRARIES

- **Transact:**
  - Transact is in Rust, with C APIs…we have been working on Go ports.
  - Use Transact submission/management code where possible across apps.
    - Ensures clean, standardized transaction handling code.
  - Experimenting on how to add app client SDK.
    - Build clean, native language libraries for apps.
    - Working Go prototype for Sawtooth (shameless plug):
      - https://github.com/taekion-org/sawtooth-client-sdk-go

Taekion

TAEKION USE OF SAWTOOTH AND THE AVAILABLE LIBRARIES

- **Ursa:**
  - Ursa has not reinvented any wheels, just made the wheel consistently round.
  - Some clients have very specific cryptography requirements:
    - Example: US Department of Defense (DoD)
  - Real need for careful abstraction and separation of all crypto components.
    - DoD has very specific requirements that are non-negotiable.
  - Ursa is a great layer in which to do this abstraction.

Taekion

WHAT IS NEEDED BY HYPERLEDGER TO SUCCEED AS THE PLACE TO COME FOR COMPOSABLE PARTS

- **(Really) Pluggable Consensus**:
  - Sawtooth and Fabric BOTH have ”pluggable” consensus.
    - Not compatible with each other!
  - Hyperledger needs a well-defined standard and interface for pluggable consensus.
  - Implementing a robust, correct version of a consensus algorithm is HARD!
    - Imagine using the same well tested and debugged code across platforms.

Taekion

WHAT IS NEEDED BY HYPERLEDGER TO SUCCEED AS THE PLACE TO COME FOR COMPOSABLE PARTS

- **Networking/Connection Management**:
  - Sawtooth uses ZMQ sockets
    - Excellent library, but still too low-level.
  - Need component that provides common communication schemes:
    - Peering/discovery
    - Gossip
    - Reliable connection management

Taekion

WHAT IS NEEDED BY HYPERLEDGER TO SUCCEED AS THE PLACE TO COME FOR COMPOSABLE PARTS

- **Block Storage**:
  - Blockchains assume replica of every block on every node.
  - Not all use-cases need this!
  - Block storage should be an abstraction.
    - Decouple from validation/consensus.
    - Allow for custom storage and replication strategies.
    - Can get fancy: erasure coding, etc.

Taekion

WHAT IS NEEDED BY HYPERLEDGER TO SUCCEED AS THE PLACE TO COME FOR COMPOSABLE PARTS

- **Key Storage:**
  - Solved problem, but no Hyperledger standard.
  - Was very important to solve early for our DoD clients.
  - Standalone, or integrated with Ursa.

Taekion

WHAT IS NEEDED BY HYPERLEDGER TO SUCCEED AS THE PLACE TO COME FOR COMPOSABLE PARTS

- **Data Exporting:**
  - Very useful to keep an “image” of current blockchain state in a DBMS or other store for complex queries.
  - Sawtooth provides “state delta” API.
    - Works very well over either ZMQ or REST.
  - API should be standardized
    - Apps which need to receive blockchain state shouldn’t have to reinvent the wheel.

Taekion

WHY NOT THE CRYPTO COMMUNITY APPROACH?

Do you you need composable components when you have:

Layer 2, Layer 3, Side Chains, State Channels, Oracles etc. ?

These approaches emerged to solve different workloads that monolithic chains could not on their own.

They require a blockchain to exit

The use of modular programming has advantages.

Function can be added and subtracted by workload

Taekion

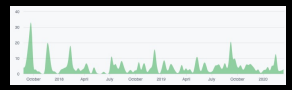
OPEN SOURCE IN BLOCKCHAINS – CAN HYPERLEDGER AGGREGATE A BLOCKCHAIN COMPONENT COMMUNITY?

The majority of top 100 blockchains no longer Open Source their code or are dead projects

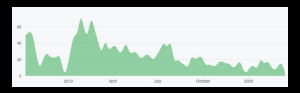
https://coincheckup.com/analysis/github

You don't have to go far from the top 100 coin listings to get to zero development in Open Source, much against the Marketing of Blockchains

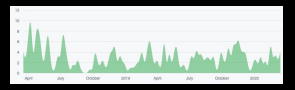
Fabric



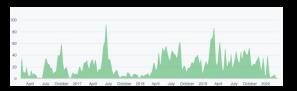
Ursa



Besu



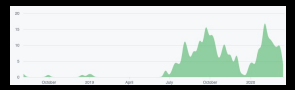
Caliper



Indy

Transact

Iroha



Avalon

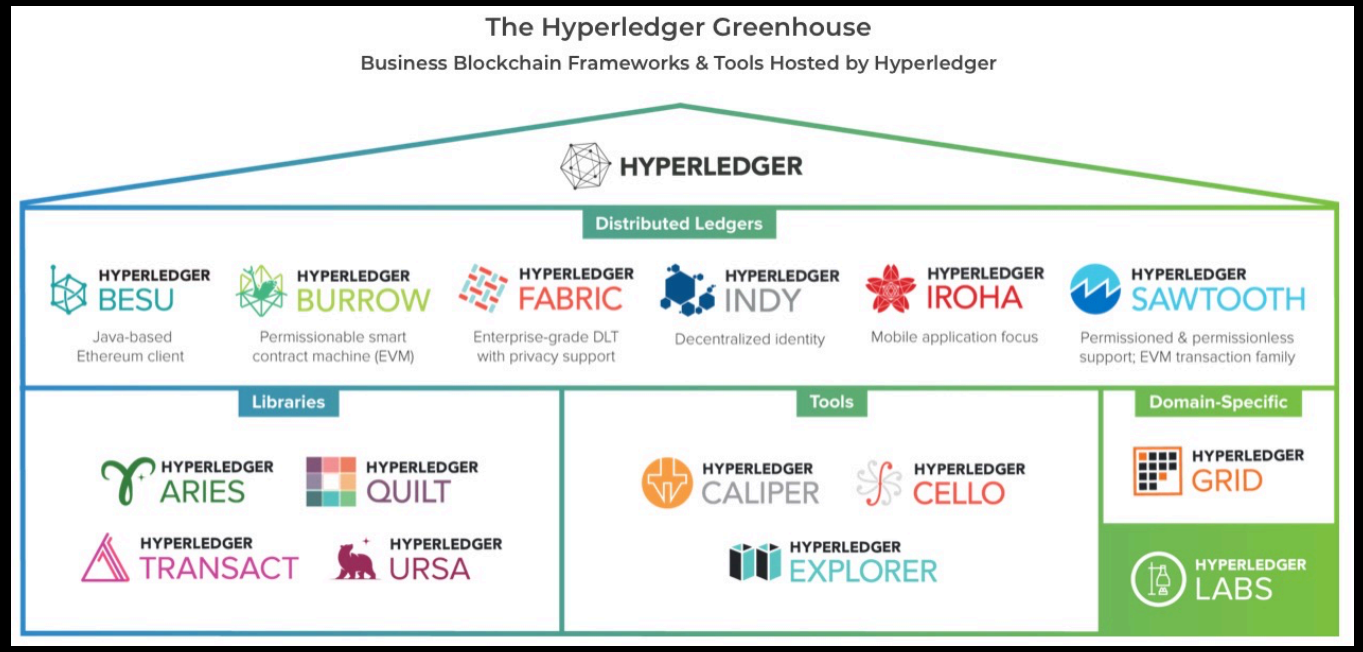
Hyperledger is the ONLY ACTIVE Open Source Community with multiple Blockchain projects including components that can be reused by any chain.

Ethereum is active but tied to the single chain.

Taekion

WITH ADDITIONAL FUNCTIONAL COMPONENTS HYPERLEDGER BECOMES THE APACHE.ORG OF BLOCKCHAINS

IN OTHER WORDS: *Should Hyperledger be the Apache.org for Blockchains or Fabric with some add-ons?*



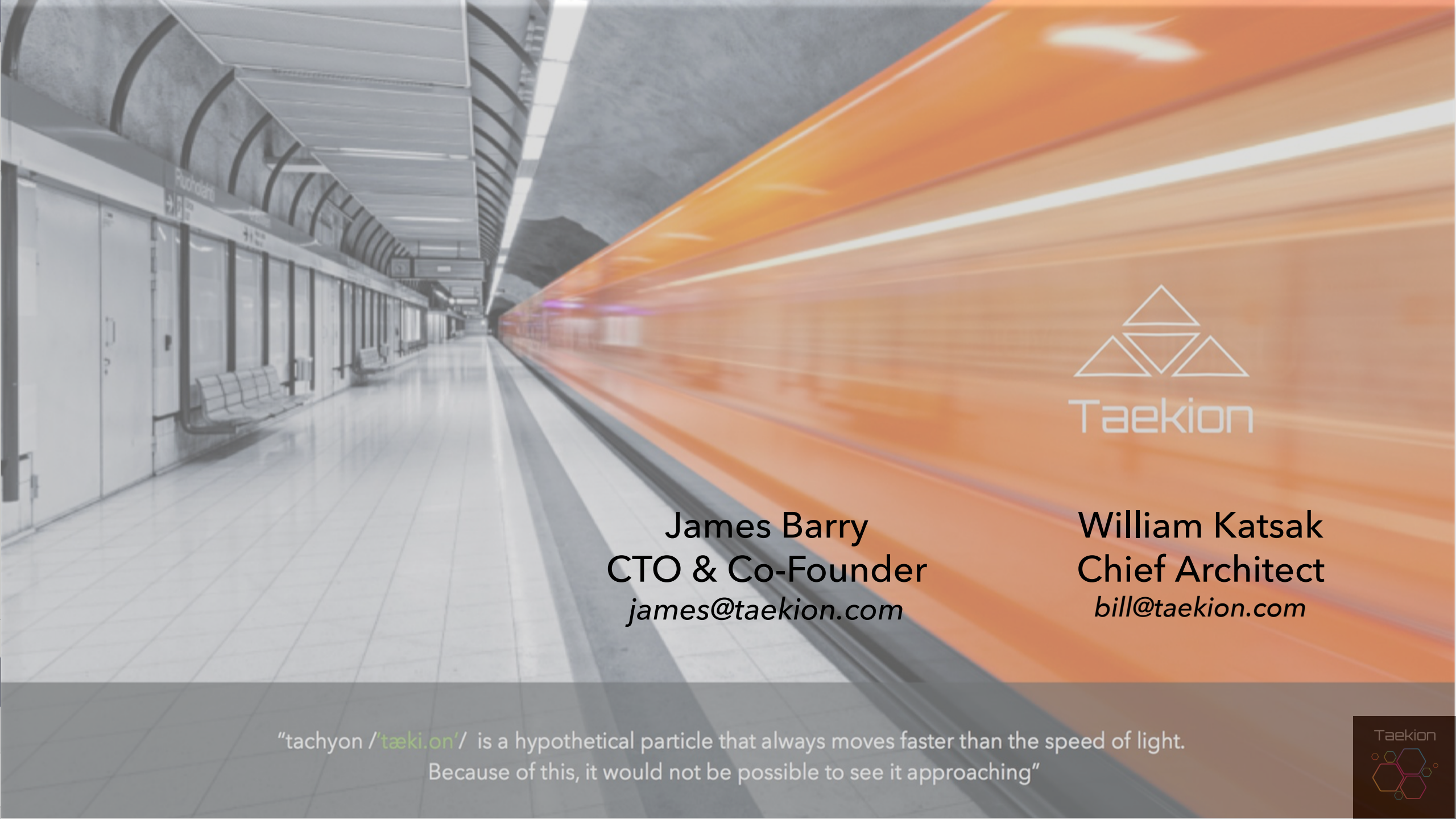
Taekion

WHERE SHOULD HYPERLEDGER TECH GO IN 

- Standardization should begin using Hyperledger Open Source as reference applications
- More specialization in components – Separate projects around Consensus/Data Storage/Transact/Ursa etc.
- General Purpose Blockchains fade away as their parts become interoperable.
- Private Chains abound, but the blockchain computational capabilities fade into the overall fabric of enterprise workflow.
- Enterprises use Public Chains only for validation - not storage or full computations
- Storage of data goes off chain in order to enable lower cost for enterprises
- Number of these blockchain parts are assembled into new solutions - blockchain components create new apps that change how work is done by workload.

This is where the real long term impact is made.

Taekion



Taekion

James Barry
CTO & Co-Founder
james@taekion.com

William Katsak
Chief Architect
bill@taekion.com

"tachyon /'tæki.on'/ is a hypothetical particle that always moves faster than the speed of light.
Because of this, it would not be possible to see it approaching"

Taekion