All On Cloud 9

Aarti Jivrajani
Abtin Bateni
Daniel Shu
Yiyang Xu

Department of Computer Science

UC SANTA BARBARA

Motivation

- Untrusted parties often need to work with each other to achieve the same goal
- Privacy concerns
- A versatile model to meet the needs of different applications

Features

- Used CAPER to build a supply chain management application
- CAPER
  - In collaborative workflows, where different players don't trust each other
  - Each player maintains their personal, confidential blockchain while respecting a log of events
- Plug and play consensus module

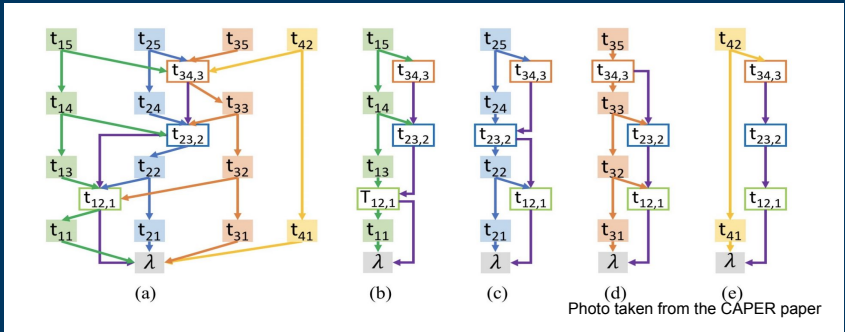
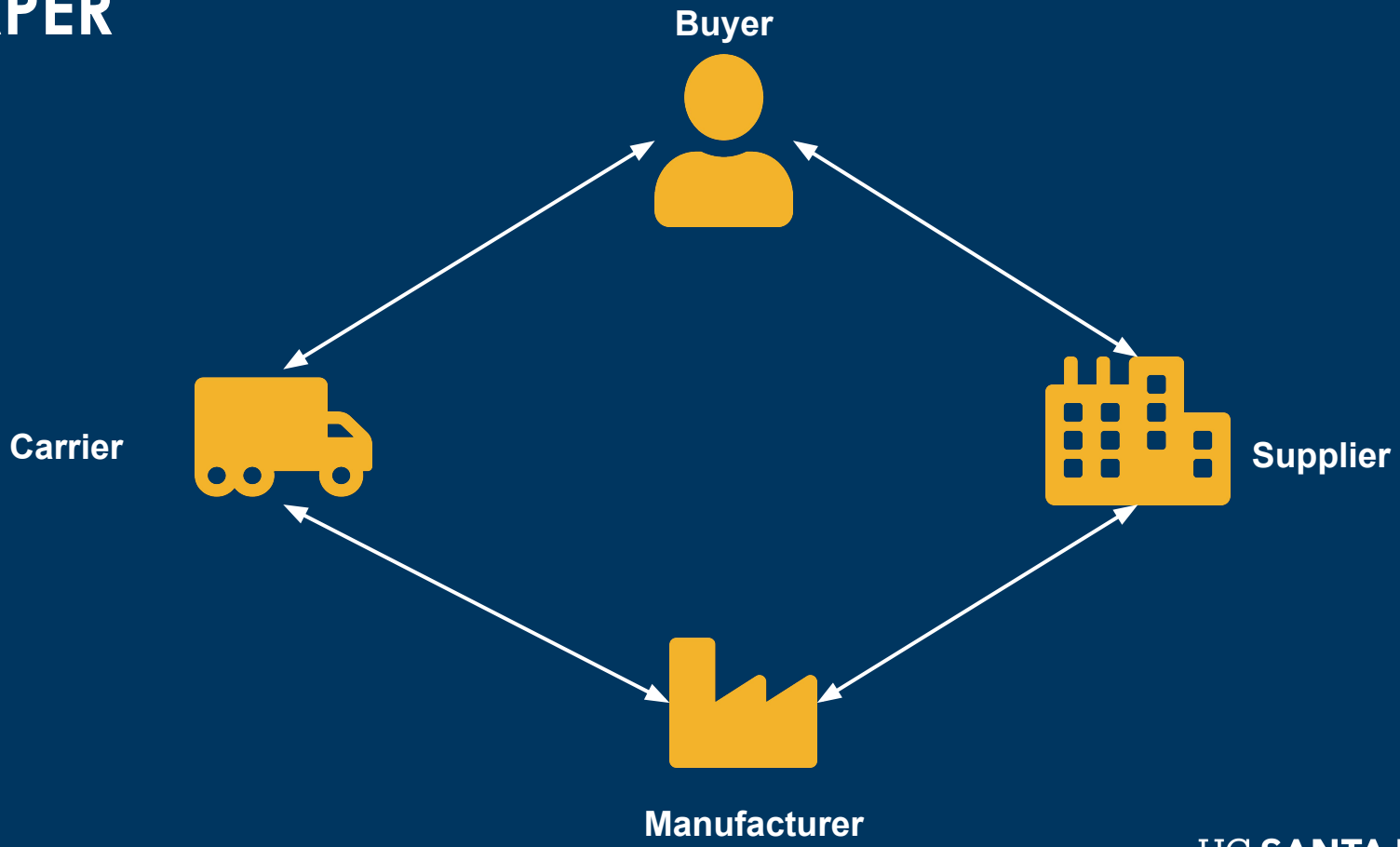
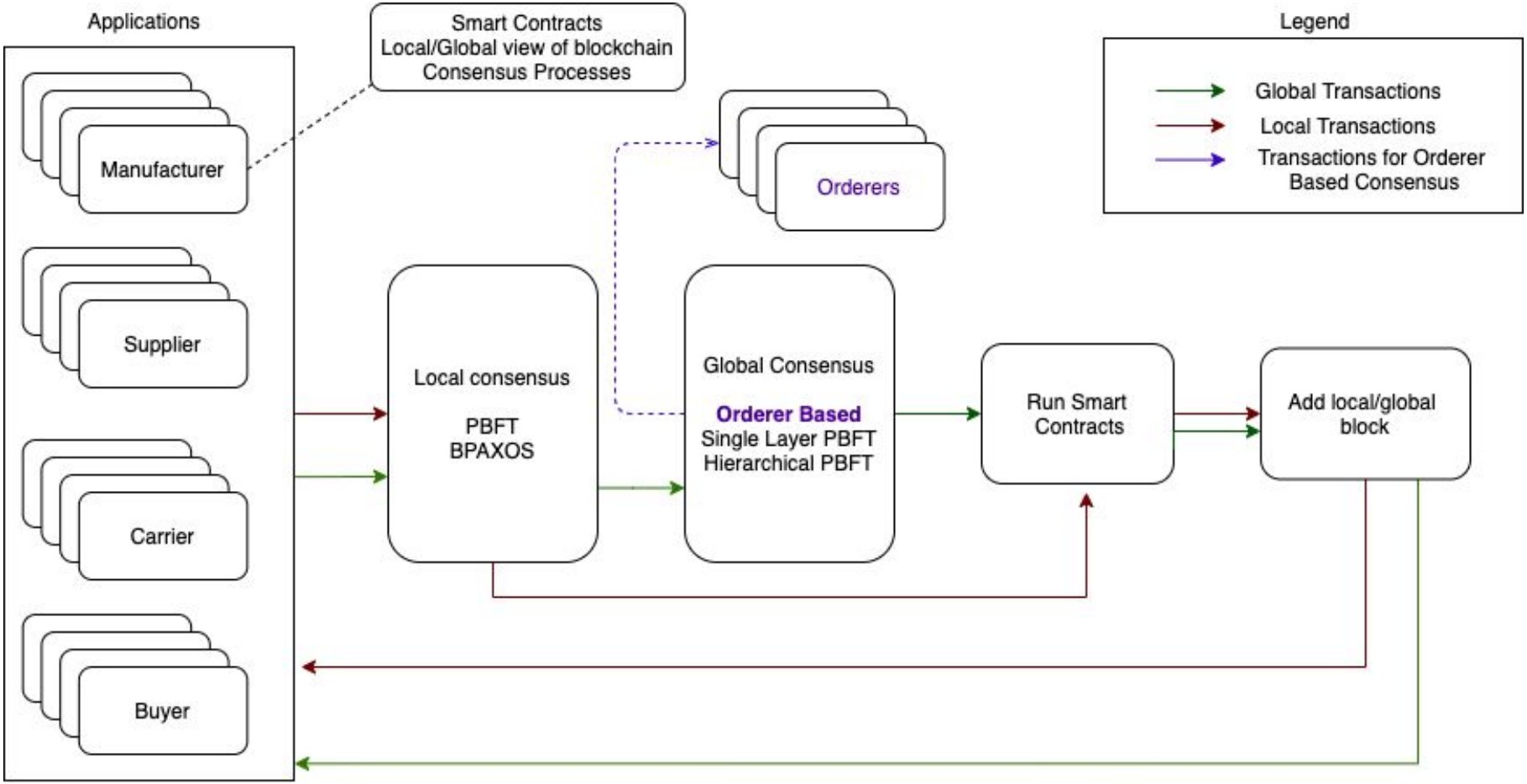


Photo taken from the CAPER paper

CAPER




Applications
Smart Contracts
Local/Global view of blockchain
Consensus Processes
Legend
Global Transactions
Local Transactions
Transactions for Orderer Based Consensus
Manufacturer
Supplier
Carrier
Buyer
Orderers
Local consensus
PBFT
BPAXOS
Global Consensus
Orderer Based
Single Layer PBFT
Hierarchical PBFT
Run Smart Contracts
Add local/global block

Communication - NATS

- Lightweight open source messaging system
  - Pub-Sub mode of communication
  - Ideal for microservices and IoT platforms(native MQTT support in Q3 2020!)

- Log-based data structure, provides an option to persist the log to disk
- TLS authentication per NATS node allowed us to move security of the system to the NATS infrastructure

Deployment Model

- Uniform runtime environment
- Dependency management

Kubernetes

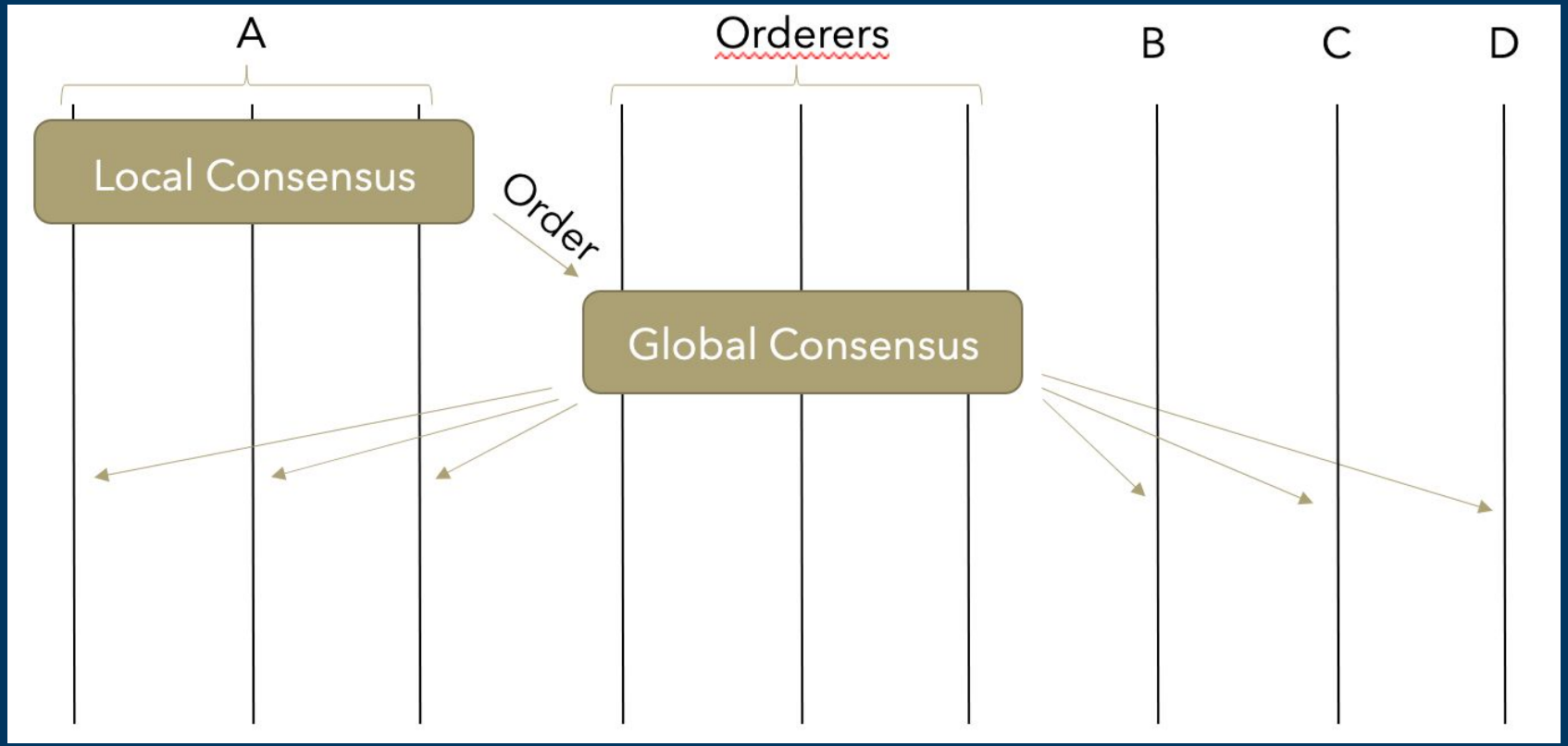
Containerized application lifecycle manager

Easy orchestration

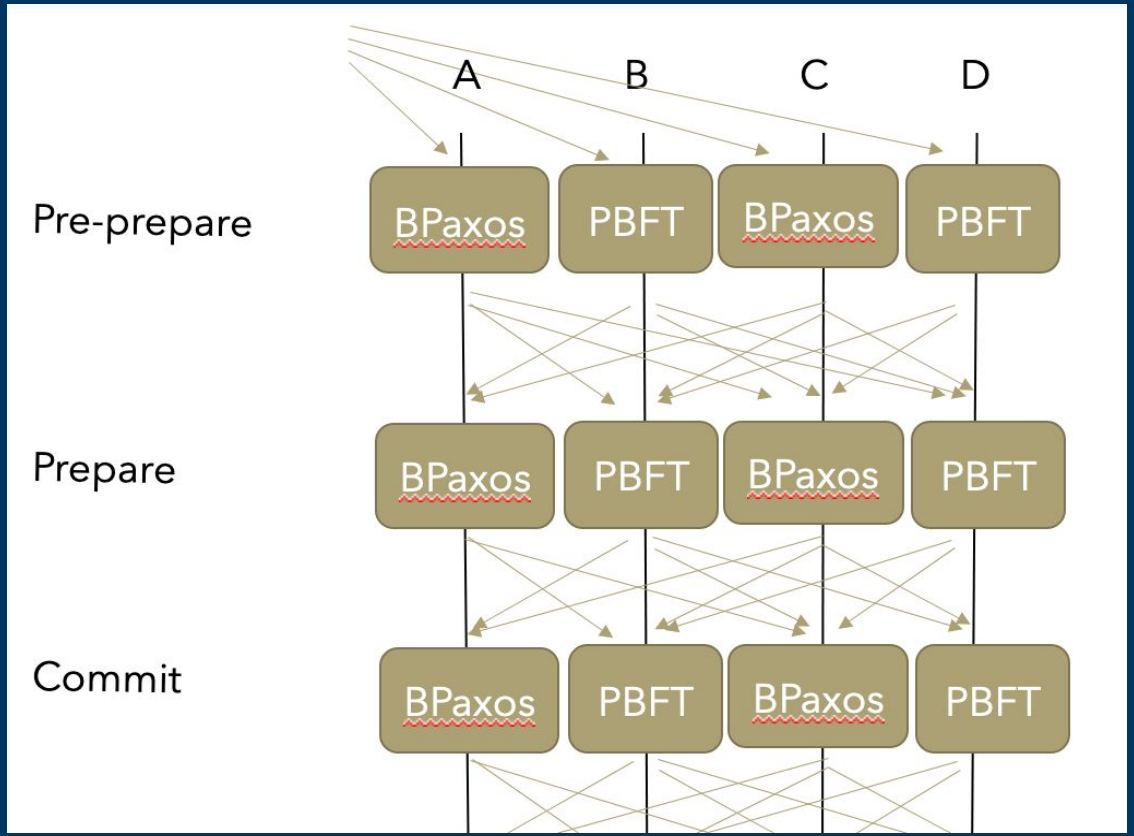
Single container per pod

Cloud deployment platform

Using a separate set of orderers



Hierarchical Consensus

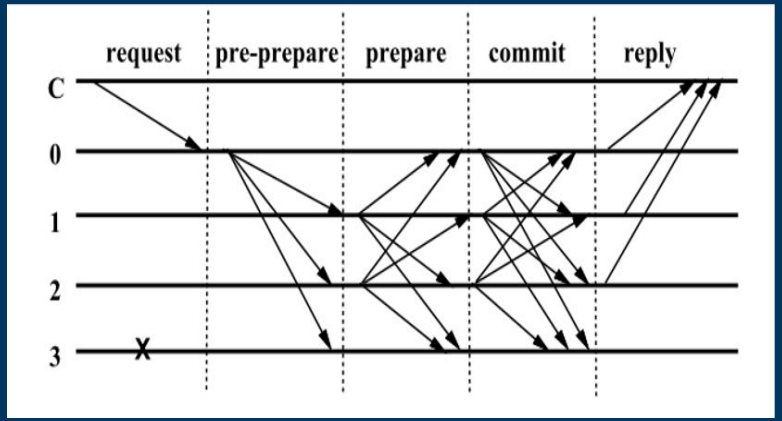


Extra features of NATS

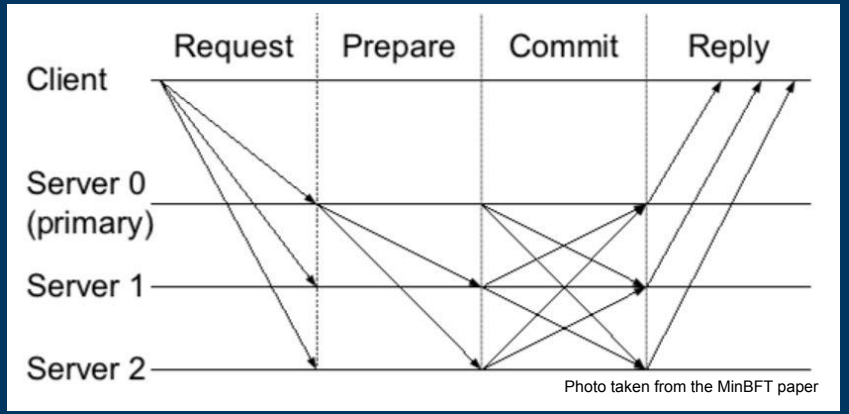
- Nats imposes a restriction to the nodes that they can't send different messages to different nodes
- A message is either sent to all or not sent at all
- Also known as using **trusted hardware**
- MinBFT is a variation of PBFT that exploits this feature

MinBFT vs. PBFT

- MinBFT has one phase less than PBFT and is essentially faster

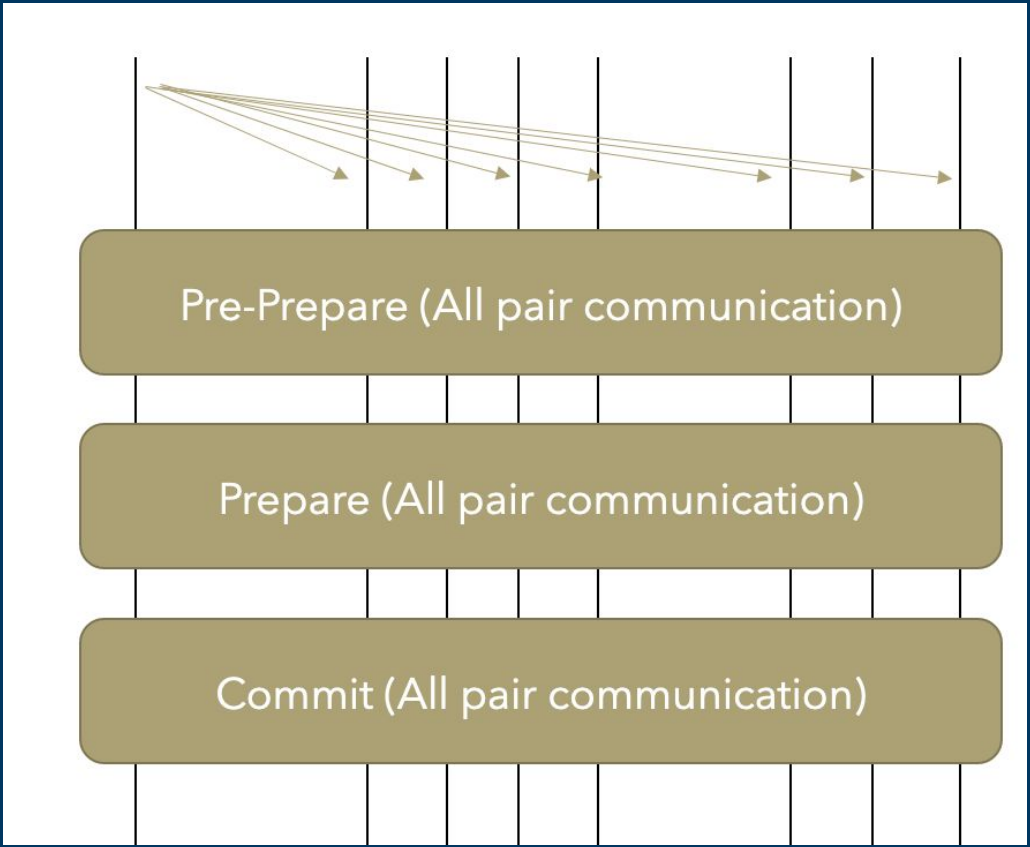


PBFT



MinBFT

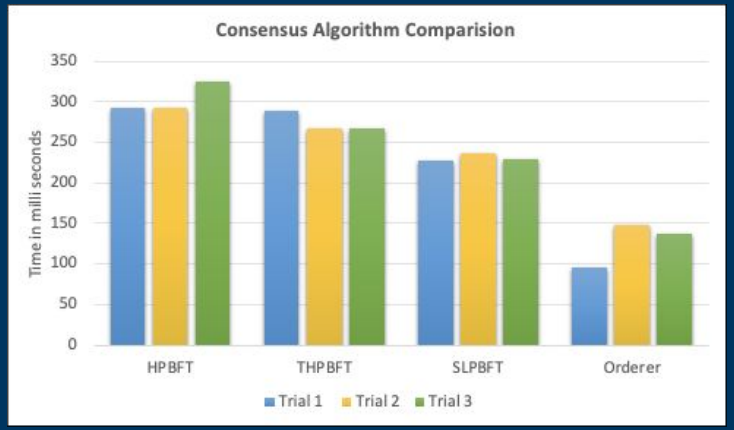
Single Layer PBFT



Comparing different algorithms

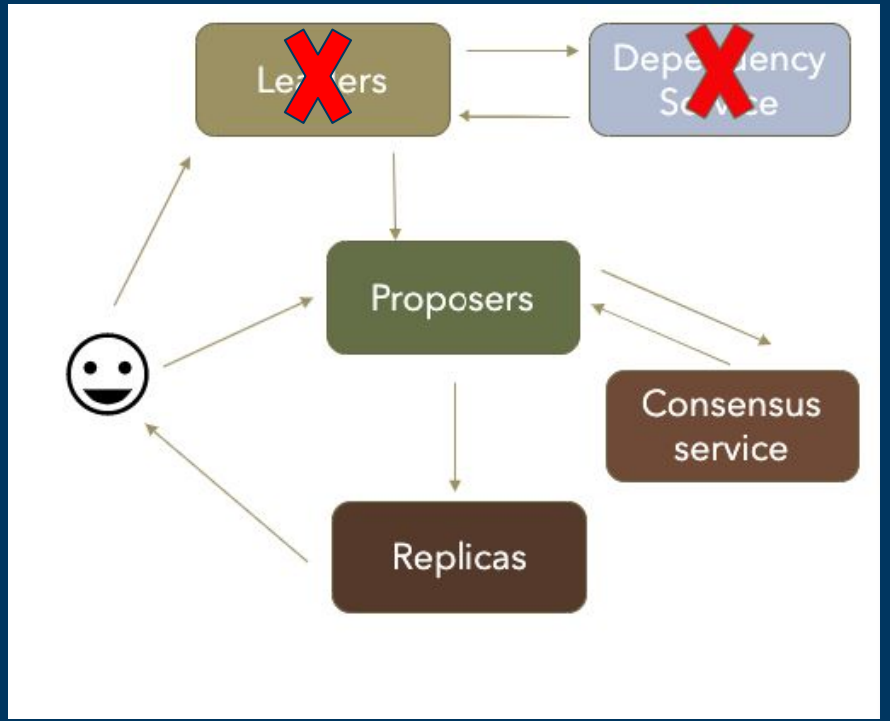
We speculate:

- Orderers are using simple BPaxos for a limited set of nodes
- Single layer PBFT does not have hierarchy
- Hierarchical MinBFT is faster than Hierarchical PBFT since a phase is removed

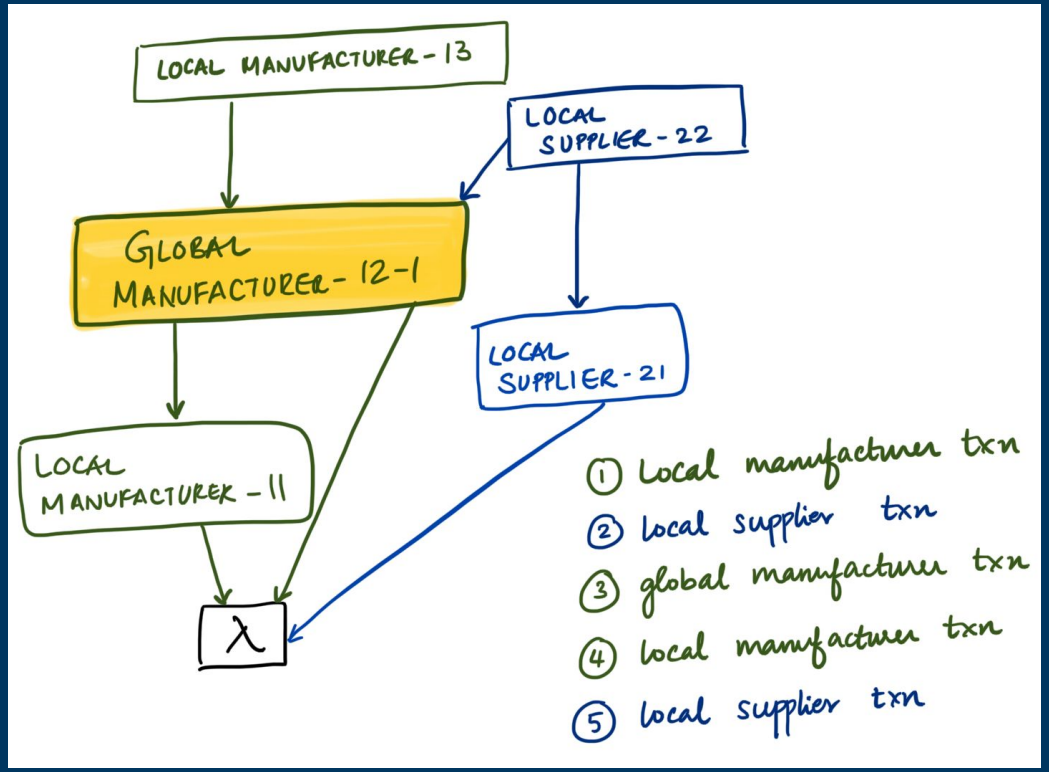


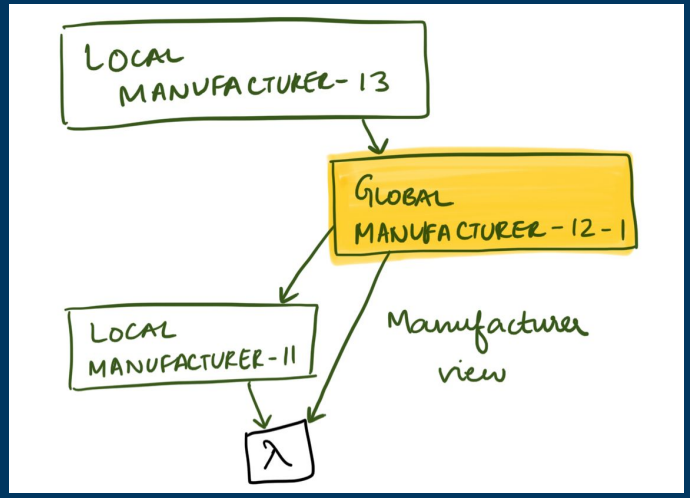
BPAXOS

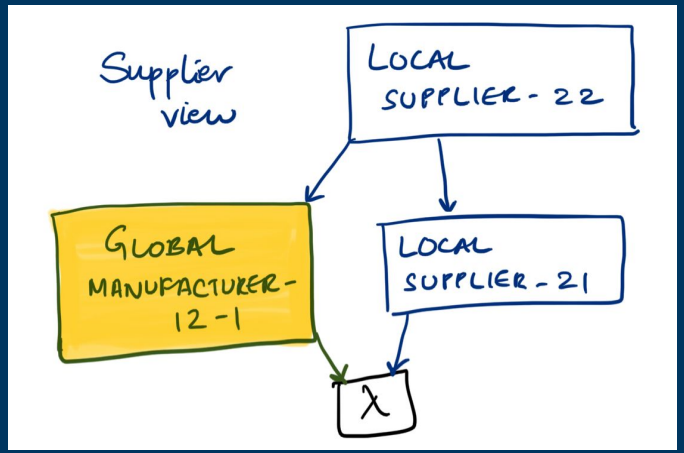
- Original BPaxos
  - Recent improvement over the original Paxos
  - Consist of 5 different services and offer simplicity and scalability due to its modularity
  - Pluggable consensus algorithm
- Our Modification
  - Remove dependency and merge leader and proposer




DEMO


LOCAL MANUFACTURER - 13
LOCAL SUPPLIER - 22
GLOBAL MANUFACTURER - 12-1
LOCAL SUPPLIER - 21
LOCAL MANUFACTURER - 11
λ
1 Local manufacturer txn
2 local supplier txn
3 global manufacturer txn
4 local manufacturer txn
5 local supplier txn

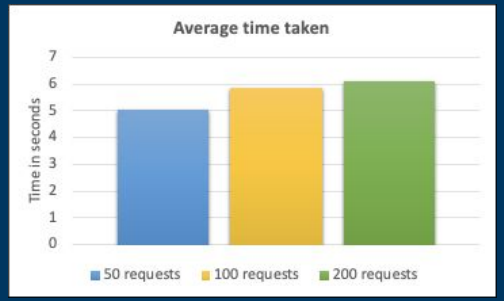

LOCAL MANUFACTURER - 13
GLOBAL MANUFACTURER - 12-1
LOCAL MANUFACTURER - 11
Manufacturer view
λ


Supplier view
LOCAL SUPPLIER - 22
GLOBAL MANUFACTURER - 12-1
LOCAL SUPPLIER - 21
λ

Scalability

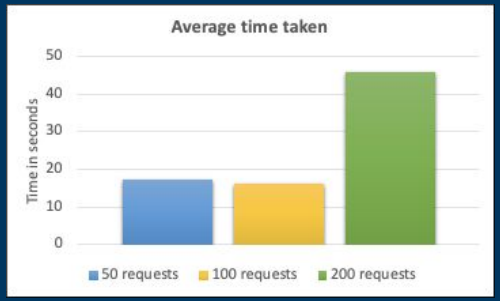
- Eucalyptus: Ubuntu Server 16.04 LTS (Xenial Xerus)
- Instance type: hi1.4xlarge
- Chose to test on Eucalyptus instead of AWS for monetary reasons

5 P
5 C
5 R

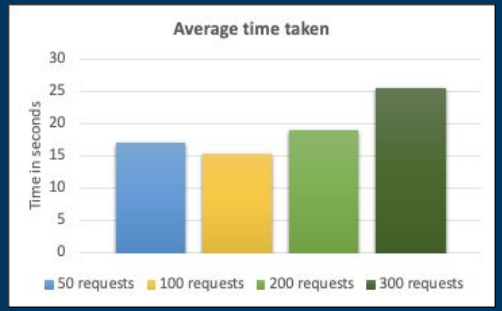


50 P
5 C
5 R

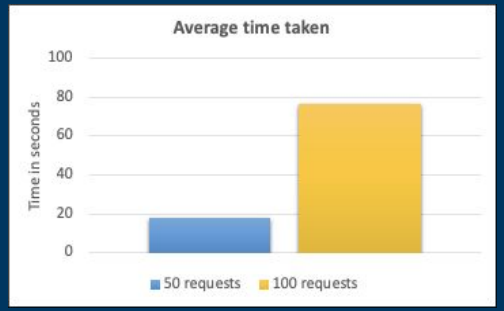
50 P
50 C
50 R



70 P
50 C
50 R



100 P
50 C
50 R



P: Proposer
C: Consensus
R: Replica

Questions?