Assignment – 1
Scaling DLT technologies.
TCDID: 22333721, Name: Pallavit Aggarwal

Paper Subject: Towards Scaling Blockchain Systems via Sharding, June 2019
Paper link: https://dl-acm-org.elib.tcd.ie/doi/pdf/10.1145/3299869.3319889
Key Contributions/Findings/Conclusions

1. Authors compare following BFT implementations: PBFT, Tendermint (variant of PBFT used by Ethermint and Cosmo), Istanbul BFT, Raft. Hardware assisted PBFT outperforms other protocols inside the TEE.
2. Hyperledger can be improved by splitting the message queue into two different channels and distributing workload. Attested Hyperledger using trusted log abstraction and AHL-Relay message aggregation are proven better in comparison to normal HL.
3. Different sharded blockchains like Elastico, OmniLedger and RapidChain are contrasted based on Unspent Transaction Output (UTXO) and the authors propose a general solution based on non-UTXO which implements 2PC and 2PL in the codebase.
4. Shard formation and reconfiguration is approached with epoch based refresh and random number signature creation which improves the security and performance from the existing implementations

Key Technology Insights

1. TEE(Trusted execution environments) like Intel SGX, Sanctum, TrustZone to provide hardware enclave or Overshadow for software-based TEE
2. Byzantine Equivocation can be eliminated by running consensus protocol inside TEE
3. Randomness Beacon Approach to generate hashed signature per epoch for security, but TEE works in a sealed-glass environment where the workings are visible to outside observer.
4. Transaction management in 2PC and 2PL

Key Insights relevant to Scalable Computing

1. Centralised ledger system has 3000-4000TPS but blockchain is limited to 3-30 TPS
2. DLT can be scaled by thinking differently about the existing approach and breaking down the individual components to adapt them to scalable infrastructure
3. Security and Consensus are two major issues preventing scaling
4. DLT can be sharded but only under computationally restricted terms, not in practical daily scenarios due to vulnerabilities.

Paper Subject: Pervasive Smart Contracts for Blockchains in IoT Systems Managed by Blockchain
Paper Link: https://dl-acm-org.elib.tcd.ie/doi/pdf/10.1145/3301403.3301405
Key Contributions/Findings/Conclusions

1. Concerns around Autonomous Execution, Heterogenous contracts, Intermittent information flow are addressed by authors using 'pervasive' smart contracts based on microserviceable modules promising flexible development.
2. 3 Microservices namely – Thing communication (addresses RESTful communication, has autonomous execution contextual logic and addresses failure in communication using AMQP and MQTT) ; Contract Database (serves as a data acess point); Interoperability support (provides support for heterogeneity) are designed as solutions to concerns.
3. Smart contracts are written in Solidity and run on the EVM. Oraclize is used to make http requests and data transfers
4. At the time of the release of this paper, smart contracts are not a core feature of IOTA

Key Technology Insights

1. Smart contracts introduced by Nick Szabo enable two IoT devices to exchange data. The Programming language such as Solidity are used to code functions for the smart contracts and run on EVM
2. IPFS (Interplanetary File System) with their content-addressed and identified hashes are ideal data sources
3. Ethereum, Hyperledger Fabric and NEO are some examples of blockchain platforms that support smart contracts.
4. A code running inside EVM cannot access outside networks and thus additional libraries like Oraclize are used.

Key Insights relevant to Scalable Computing

1. IoT with a large number of heterogenous devices with sensitive data can benefit from the distributed decentralized nature of blockchains.
2. IoT smart contracts can be approached with a microservice architecture and mindset to tackle scalability
3. It is challenging to support inter-device communication in real-time and overcoming intermittent information flow due to EVM silo-boxing from outside world.
4. More power is required to run a peer-to-peer network to host this setup. Normal IoT devices are currently not that powerful in terms of compute and resilience.