DFINITY

The Internet Computer and its networks

Yotam, Yvonne-Anne, Rüdiger, April, 2022

We are hiring: dfinity.org/careers

Agenda

1) What is the IC?

2) What are its networking patterns and requirements?

3) Show me the numbers!

4) Q&A

DFINITY

What is the Internet Computer?

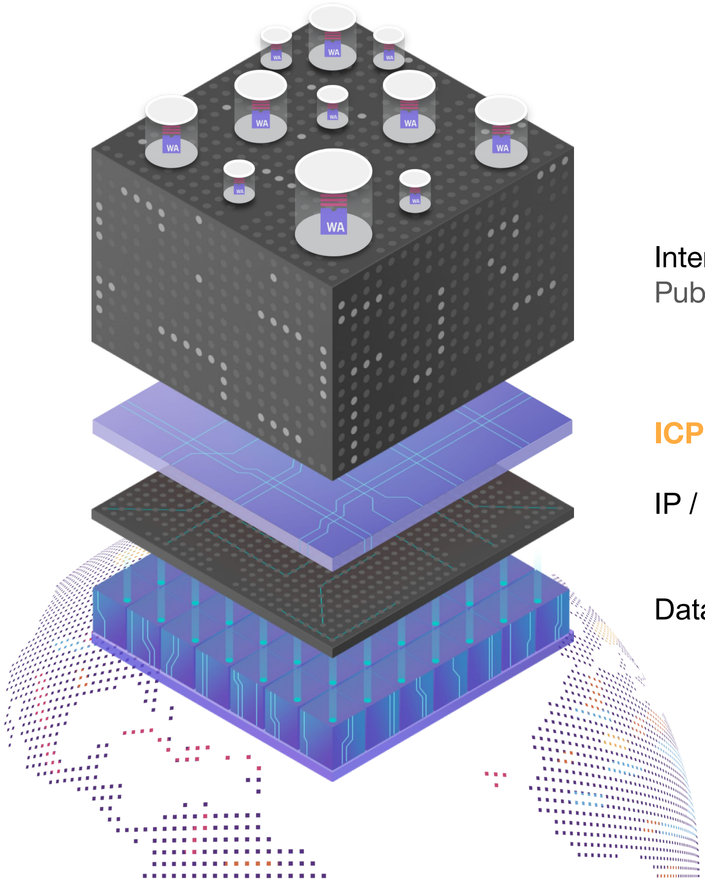
What is the Internet Computer?

Platform to run **any computation**, using blockchain technology for decentralisation and security

Internet Computer Protocol (ICP)

Coordination of nodes in independent data centers, jointly performing any computation for anyone

- Create Internet Computer blockchains
- Guarantee safety and liveness of smart contract execution despite Byzantine participants



Internet Computer
Public cyberspace

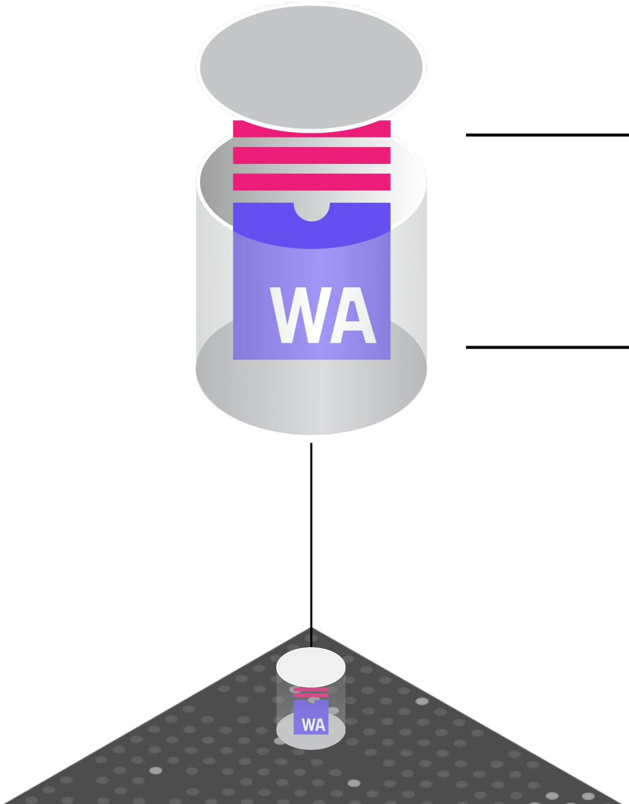
ICP

IP / Internet

Data Centers

DFINITY

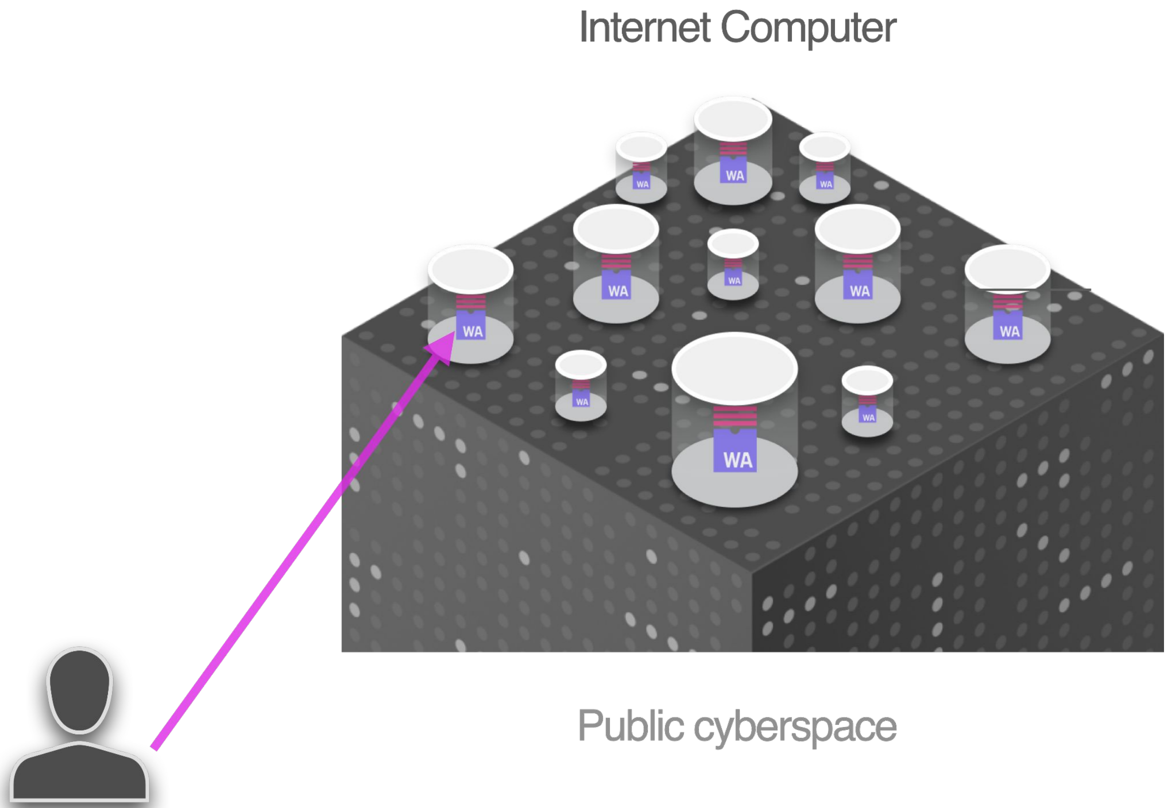
Canister Smart Contracts



Data: Memory pages

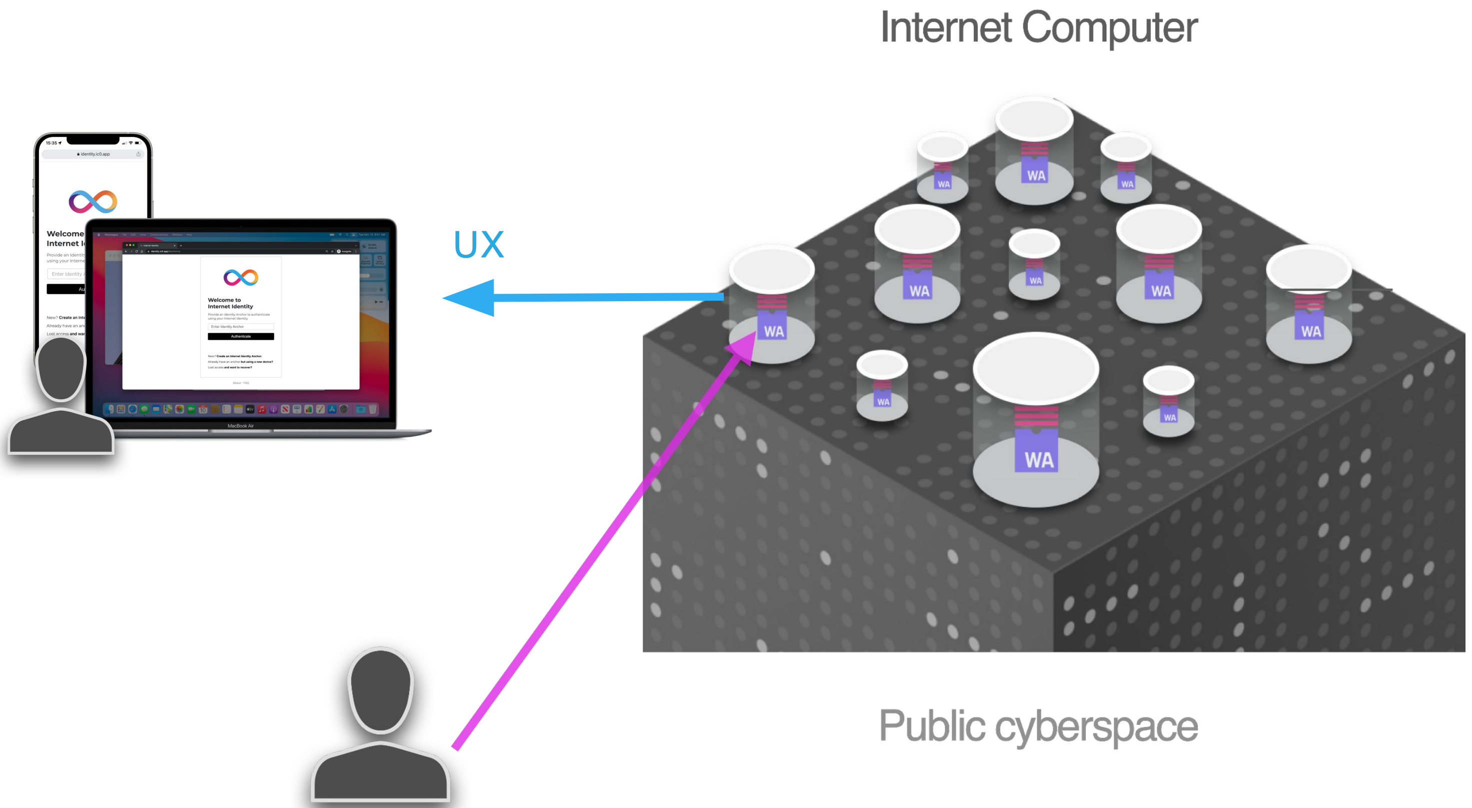
Code: WebAssembly bytecode

Developers and users interact directly with Canisters



DFINITY

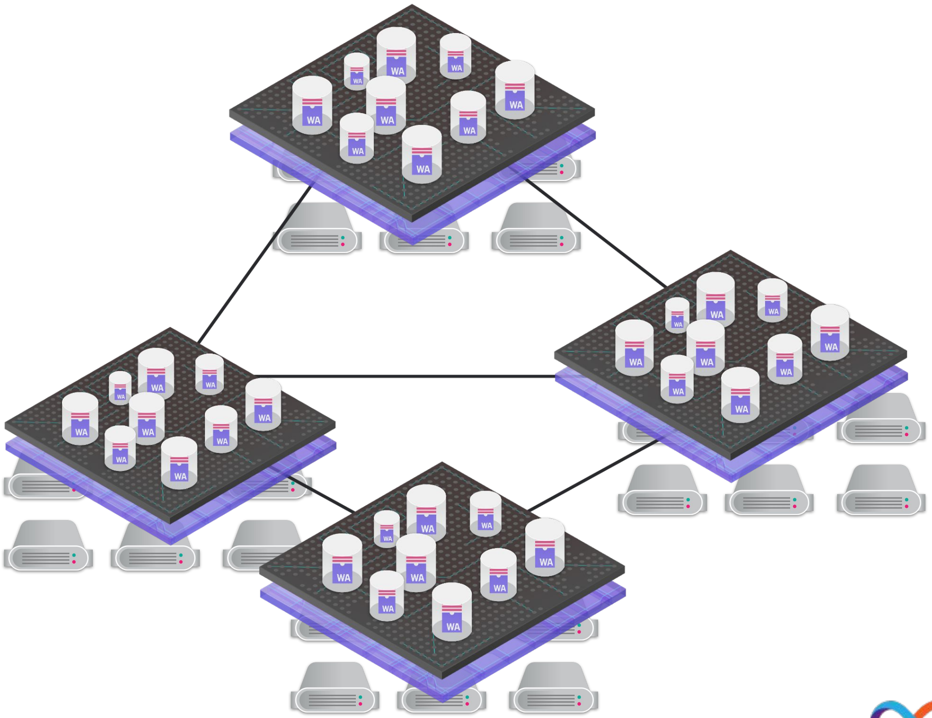
Developers and users interact directly with Canisters



Scalability: Nodes and Subnets

Nodes are partitioned into subnets

Canister smart contracts are assigned to different subnets



DFINITY

Scalability: Nodes and Subnets

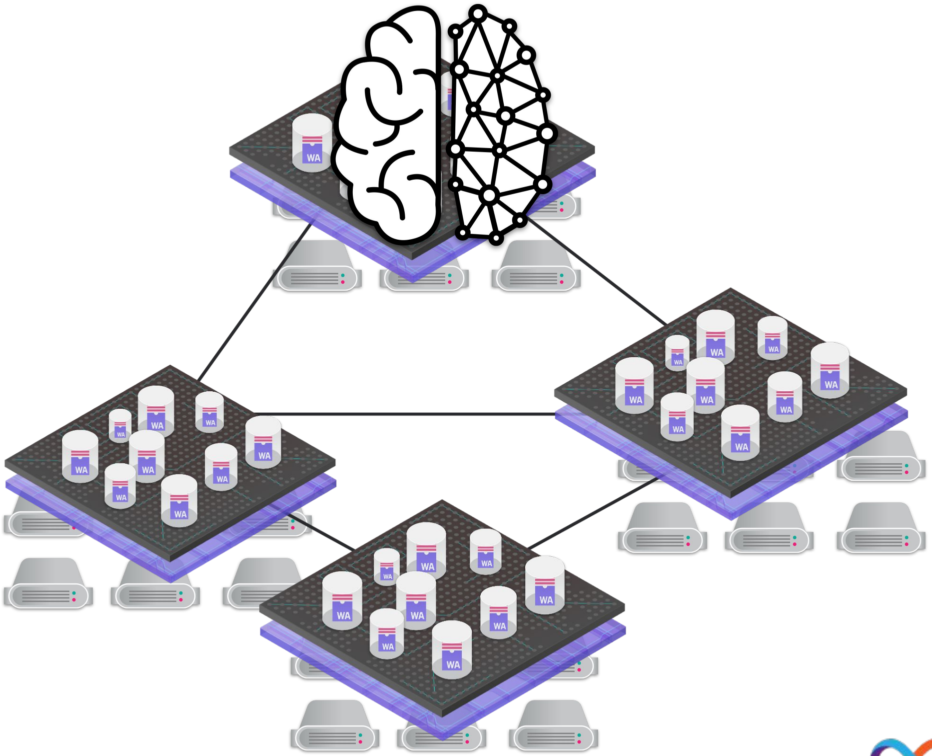
Nodes are partitioned into subnets

Canister smart contracts are assigned to different subnets

One subnet is special: it host the **Network Nervous System (NNS)** canisters which govern the IC

ICP token holders vote on

- Creation of new subnets
- Upgrades to new protocol version
- Replacement of nodes
- …



DFINITY

State Machine Replication

State:

- canisters and their queues

Inputs:

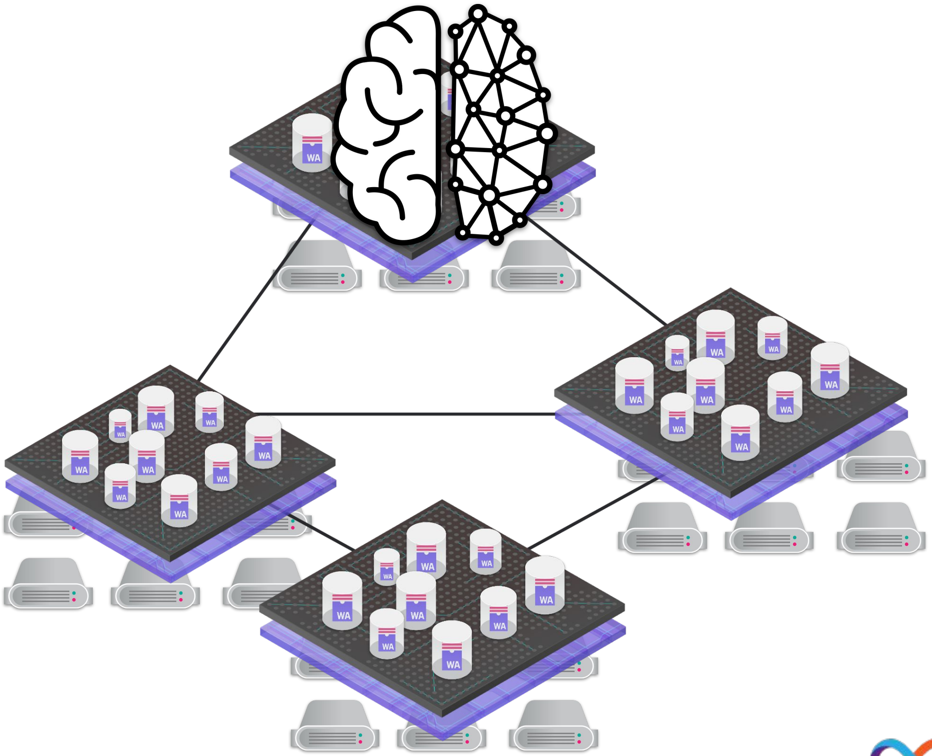
- new canisters to be installed,
- messages from users and other canisters

Outputs:

- responses to users and other canisters

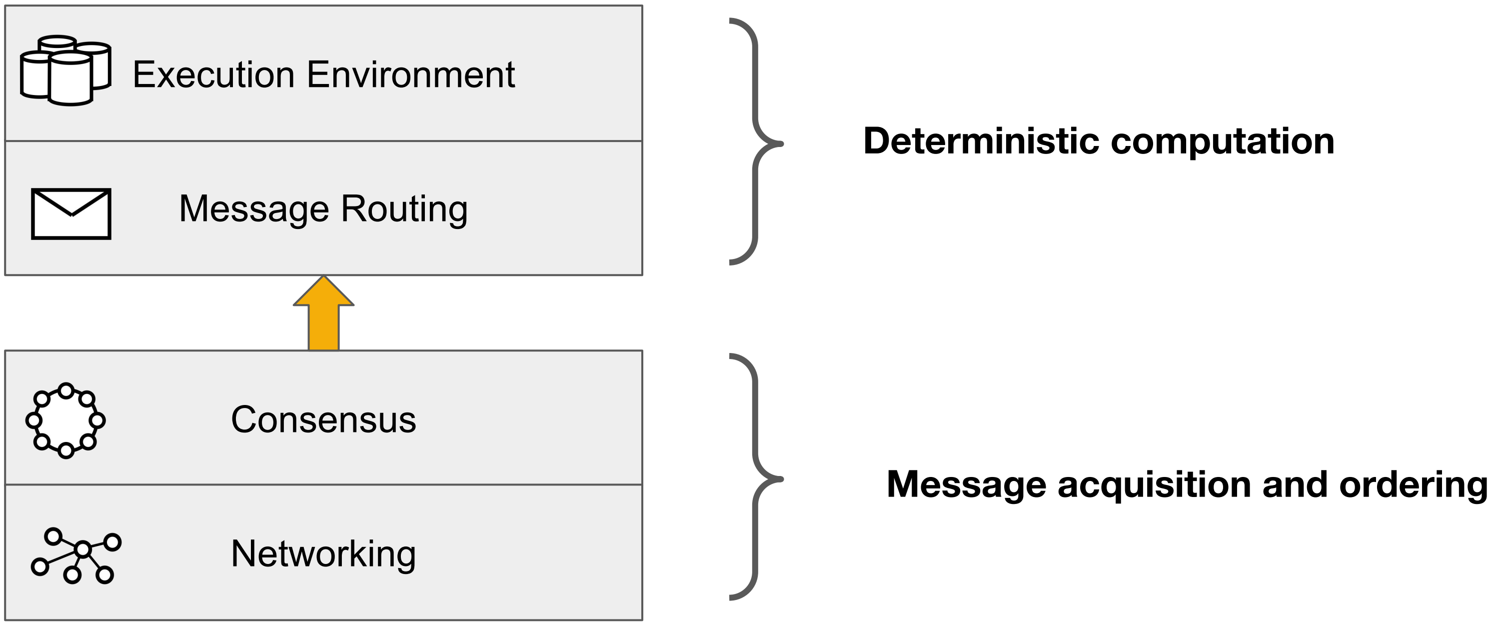
Transition function:

- message routing and scheduling
- canister code



DFINITY

ICP Layers


Execution Environment
Message Routing
Consensus
Networking
Deterministic computation
Message acquisition and ordering

Comparison with other Blockchain Systems

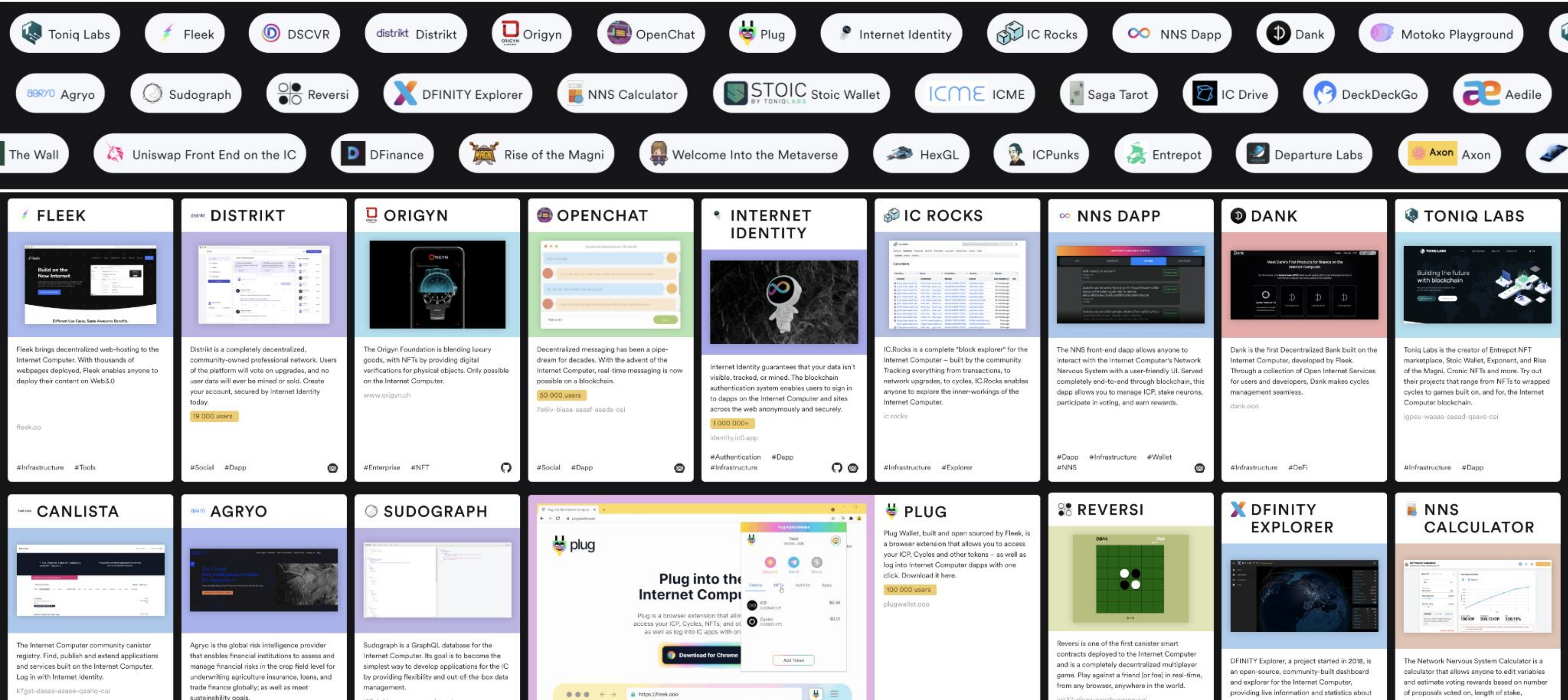
coincodex

Layer-1 Performance Comparison

	Ethereum	Cardano	Solana	Avalanche	Algorand	Internet Computer
Transaction Speed	15-20 TPS	2 TPS	2,000-3,000 TPS	4,500 TPS	20 TPS	11,500 TPS 250,000 QPS
Transaction Finality	14 minutes	10-60 minutes	21-46 seconds	2-3 seconds	4-5 seconds	1 second
Scalability	Not very scalable	Not very scalable	Not very scalable	Not very scalable	More scalability	Indefinite scalability
Node Count	6,000 nodes	3,173 nodes	1,603 nodes	1,243 nodes	1,997 nodes	443 nodes
Storage Costs	$73,000,000 / GB	Inadequate data storage	$1,000,000 / GB	$988,000 / GB	IPFS off-chain storage	$5 / GB
Cloud Service Dependency	70% of nodes run on AWS	Unclear how many are cloud	Most nodes run on cloud	Unclear how many are cloud	Most nodes run on cloud	Independent data centers

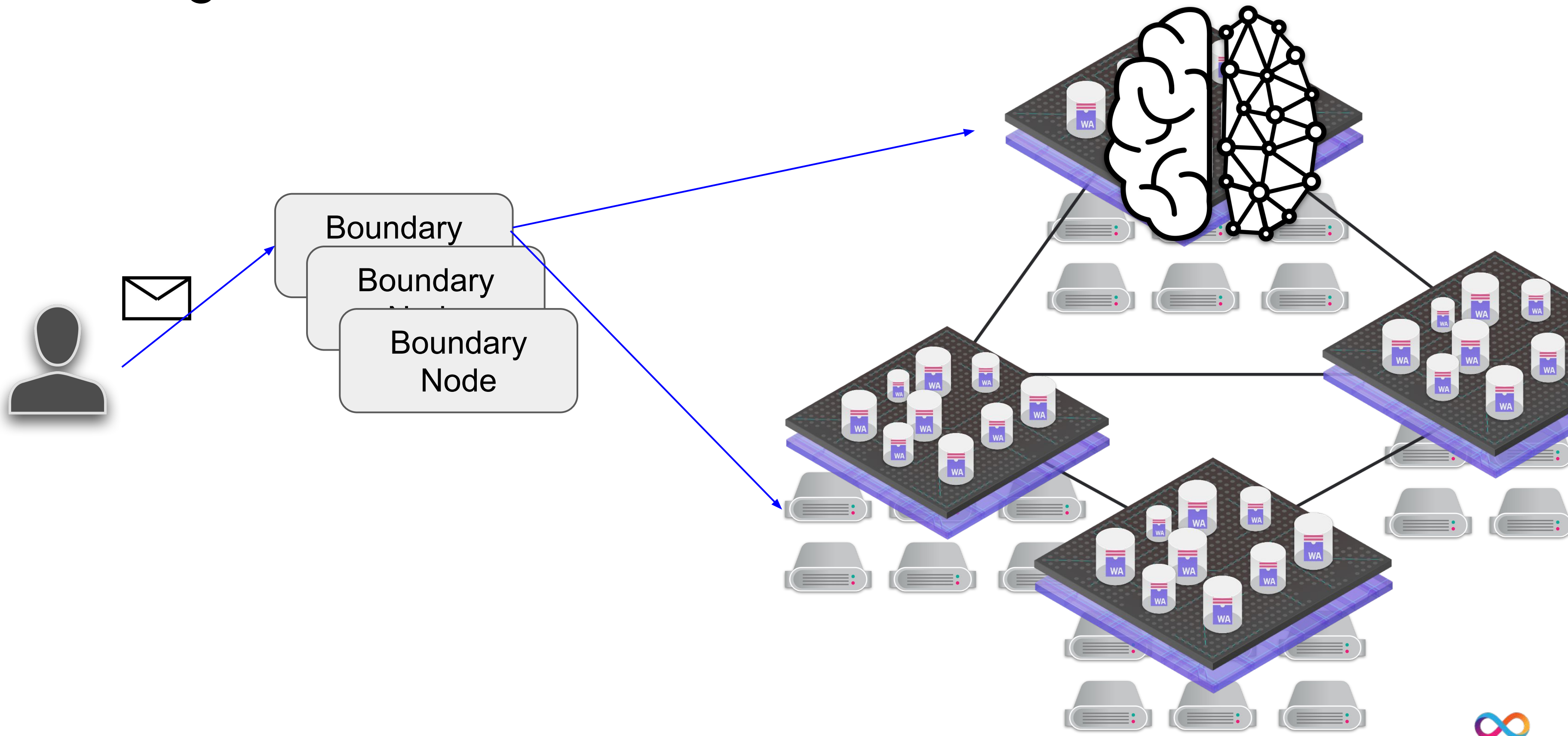
https://coincodex.com/article/14198/layer-1-performance-comparing-6-leading-blockchains/

Fast Growing Ecosystem

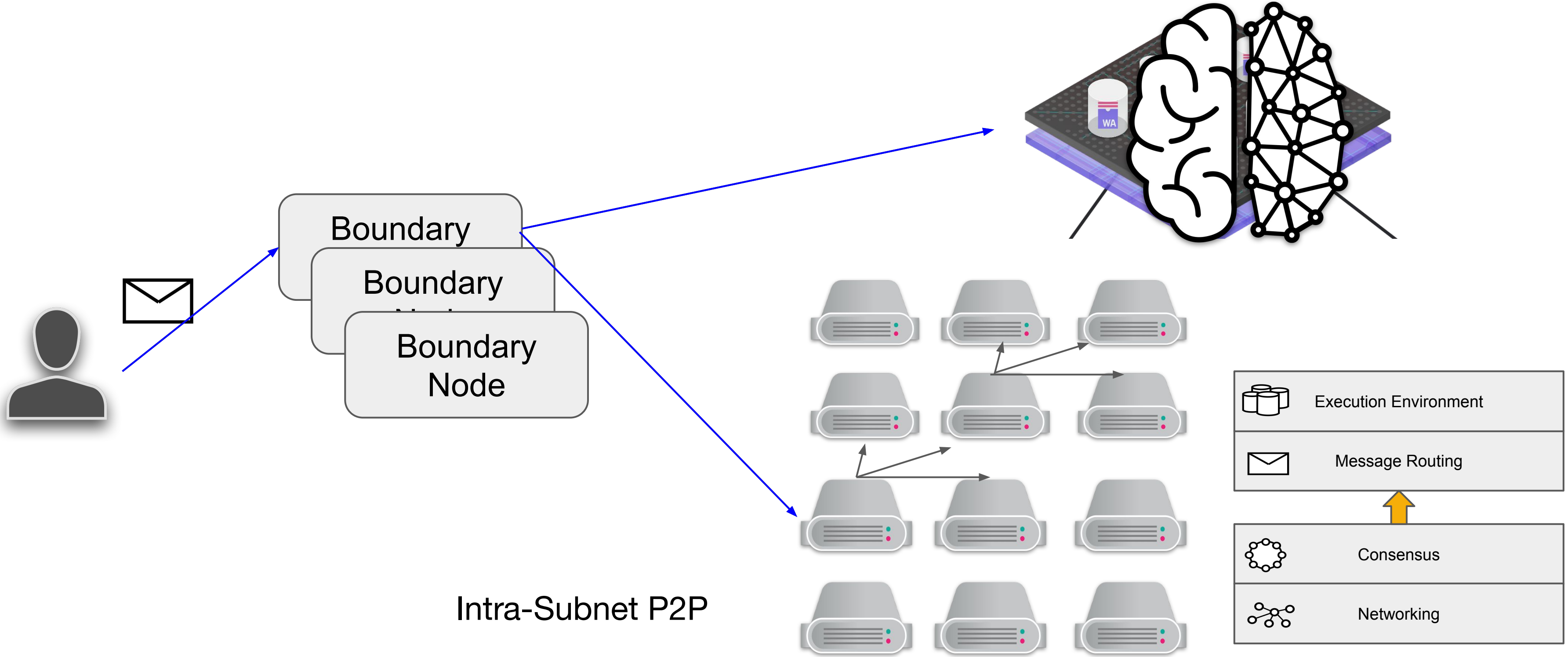


IC Networking

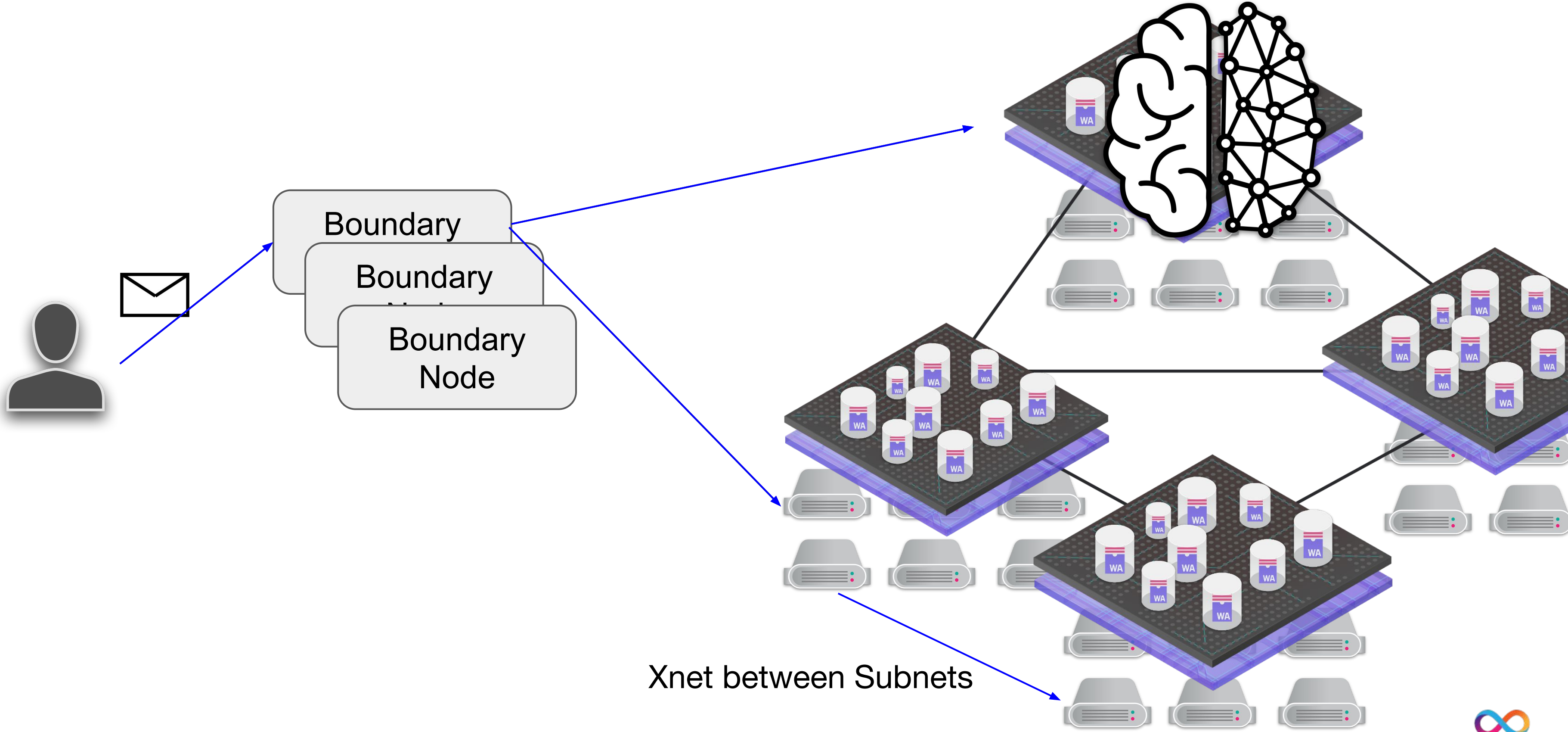
Following a canister call



Following a canister call



Following a canister call



Requirements 1/2

- **Bounded-time/eventual delivery despite Byzantine faults**

Up to a certain maximum volume of valid artifacts that are not dropped by any honest node reaches all honest nodes in bounded time/eventually despite attacks (under certain network assumptions).

- **Reserved resources for different components/peers**

Memory/bandwidth/CPU guarantees for different components and peers

- **Prioritization for different artifacts**

Not all artifacts are equal, different priorities depending on attributes (e.g., type, size, round,...). Priorities change over time.

Requirements 2/2

- **High efficiency**

High throughput is more important than low latency

Avoid duplicates: don’t waste bandwidth downloading same artifact “too many times”

- **DOS/SPAM resilience**

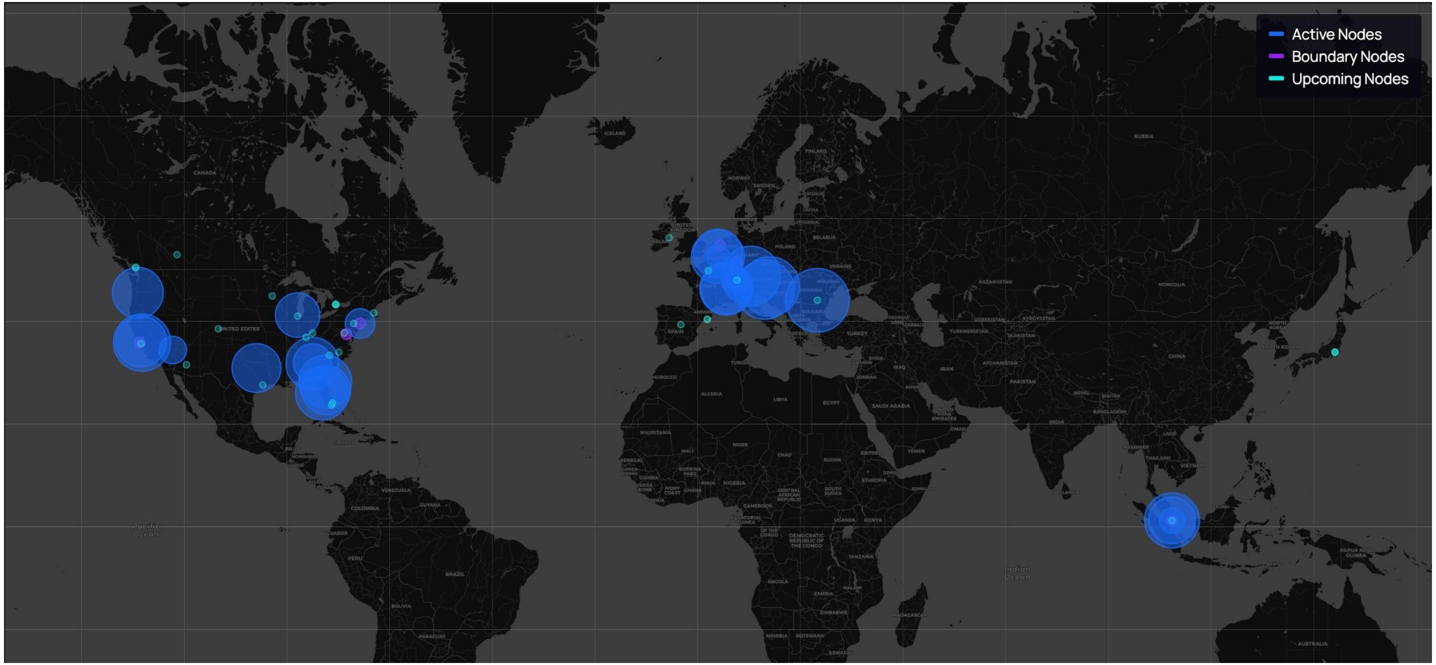
Bad participants cannot prevent progress.

- **Low accessibility requirements for users**

Support browser and IPv4 access

Networking of the IC

- **Geographically distributed**: datacenters all over the world



Networking of the IC

- **Geographically distributed**: datacenters all over the world
- **Decentralized**: a subnet is composed of nodes in different datacenters
  → Some nodes in the same subnet may be very far apart
  → Independent node providers with different skills and DC contracts
  → Communication over public internet
  - High latencies possible
  - Many transient network failures
- **Secure**: a subnet should make progress even if up to ⅓ of the nodes are faulty / malicious
  → We can't trust specific nodes (e.g., geographically close by)
  → Even nodes in the same subnet should not trust each other

Intra-Subnet P2P Networking

- Peer-to-peer network of nodes
  - Gossip protocol for artifact distribution
    - Advert - Request - Response
  - Eventual / bounded time delivery with priorities (~reliable broadcast optimized for Consensus)

- Untrusted communication
  - TLS / TCP to all nodes in the subnet, certificates in NNS
  - Authenticity and integrity of artifacts can be verified by higher layers
  - Nodes can still do evil

Xnet Inter-Subnet Networking

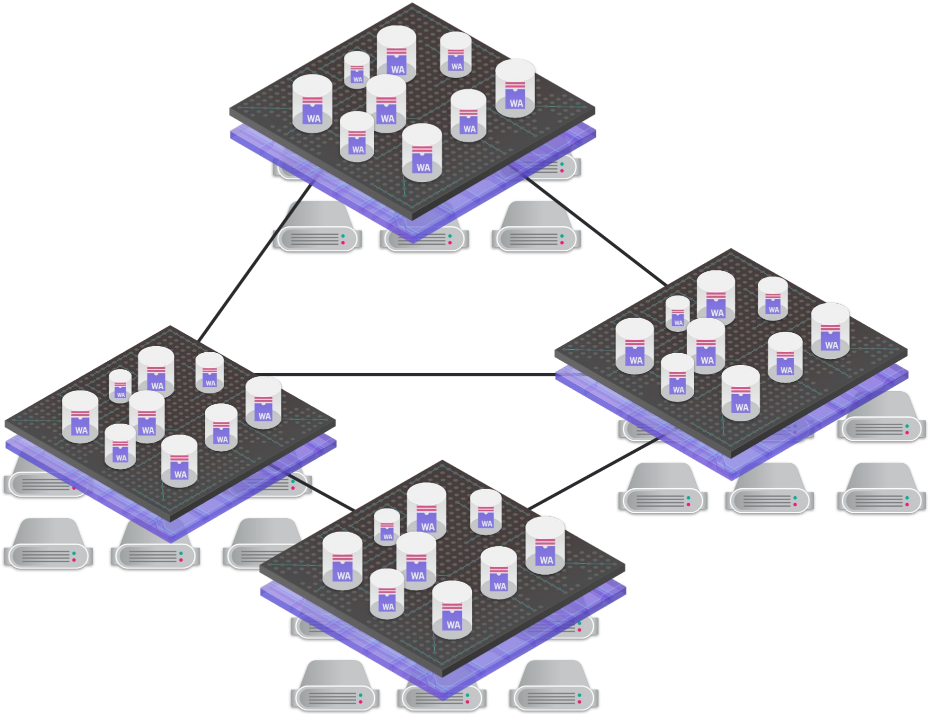
- Canisters on one subnet can send messages to canisters on other subnets, called “cross-net communication” (or Xnet)
- Currently this is done quite naively, where any node on one subnet can fetch messages from any other node on the other subnet with a HTTPS request
- We can probably improve this on several aspects:
  - Scalability: decide which nodes connect to which, and when
  - Performance: leverage the fact that some nodes in both subnets are close to each other (content is signed by the subnet, so we do not need to trust a specific node up to some extent)

Numbers…

The IC in Current Numbers

<u>Application Layer:</u>

- 60K+ canisters (smart contracts/dapps)
- > 2 Mio registered identities
- ~1TB total state (and counting…)



https://dashboard.internetcomputer.org/

DFINITY

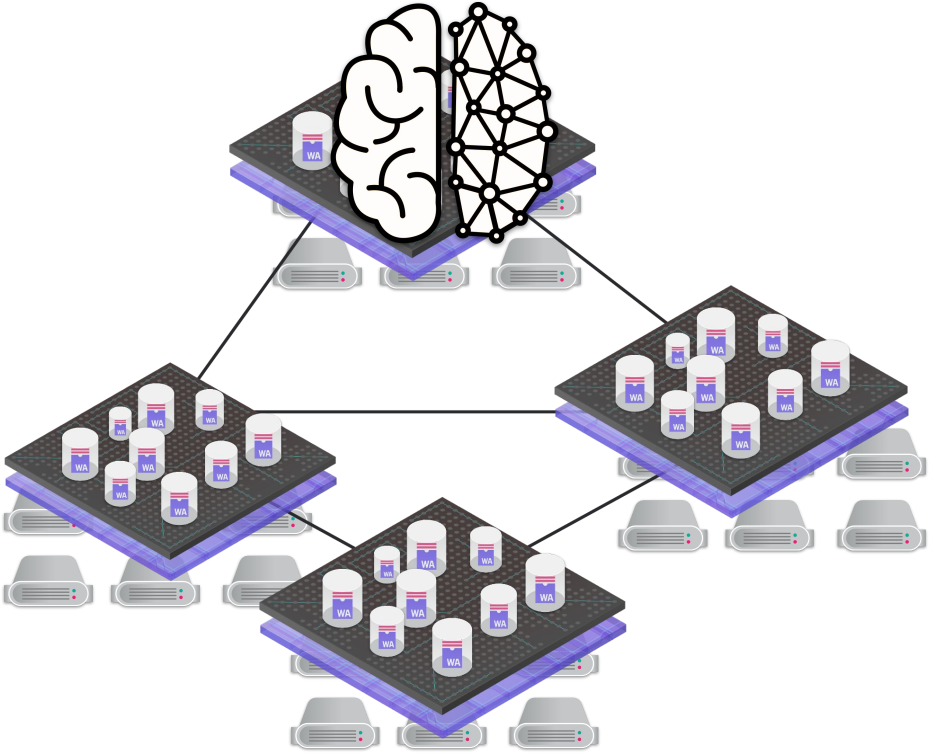
The IC in Current Numbers

<u>Application Layer:</u>

- 60K+ canisters (smart contracts/dapps)
- > 2 Mio registered identities
- ~1TB total state (and counting…)

<u>Governance:</u>

- So far:
  - 56K+ NNS proposals
  - 3.4M+ ICP transactions



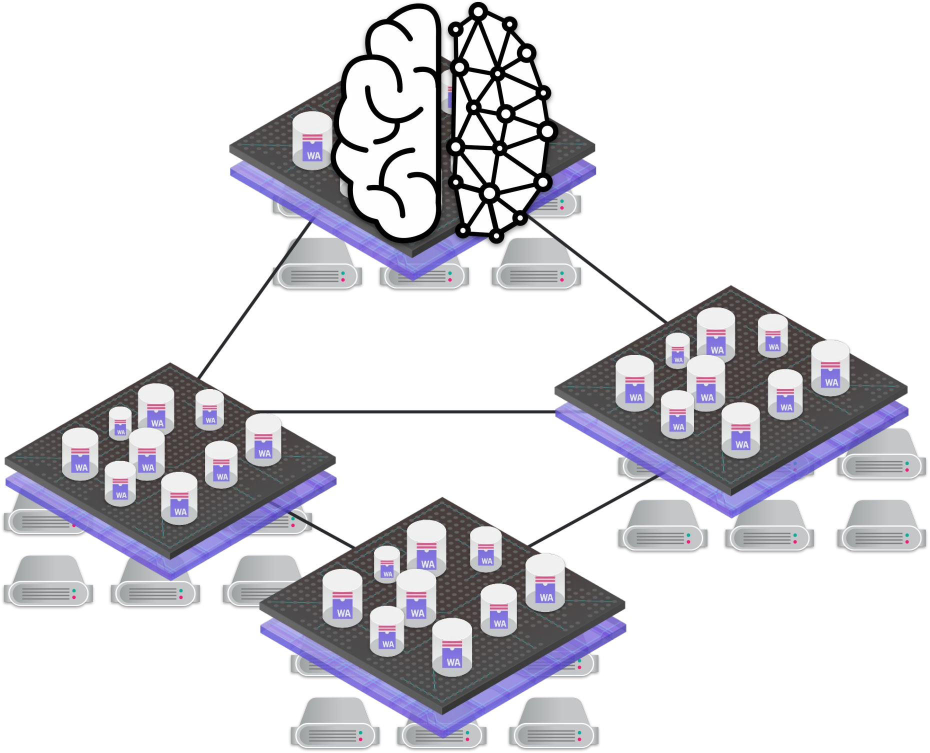
https://dashboard.internetcomputer.org/

DFINITY

The IC in Current Numbers

Consensus

- 758M+ blocks created
- ~34 blocks per second
- ~2800 transactions per second



https://dashboard.internetcomputer.org/

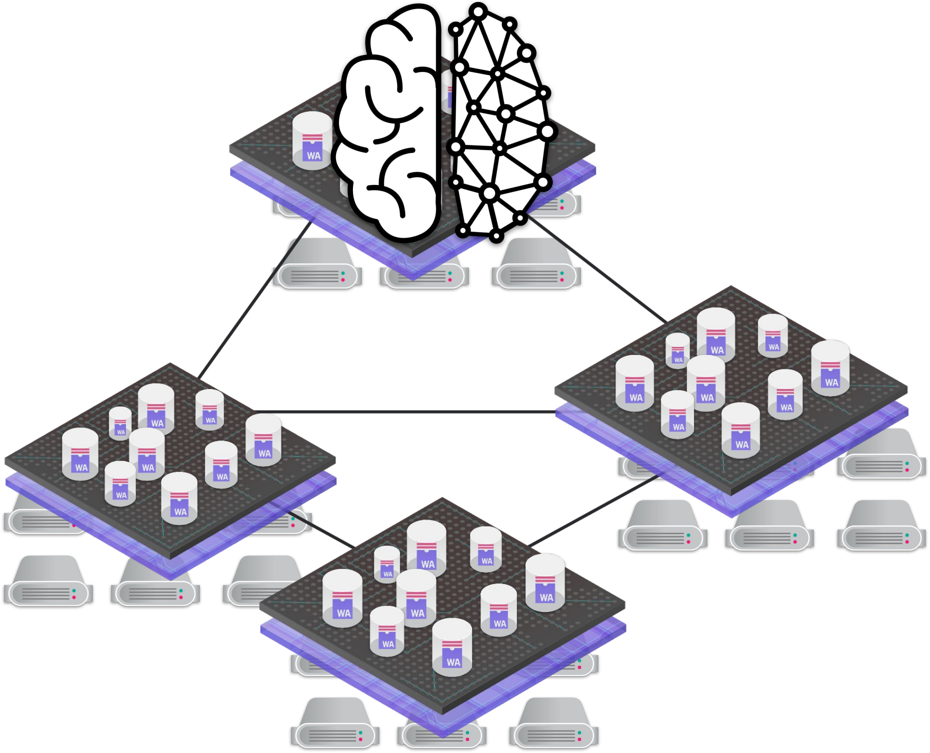
DFINITY

The IC in Current Numbers

Network Layer:

- 477 nodes
  - From 54 node providers
- 33 subnets
  - 40 nodes in NNS subnet
  - 13 nodes in App subnets

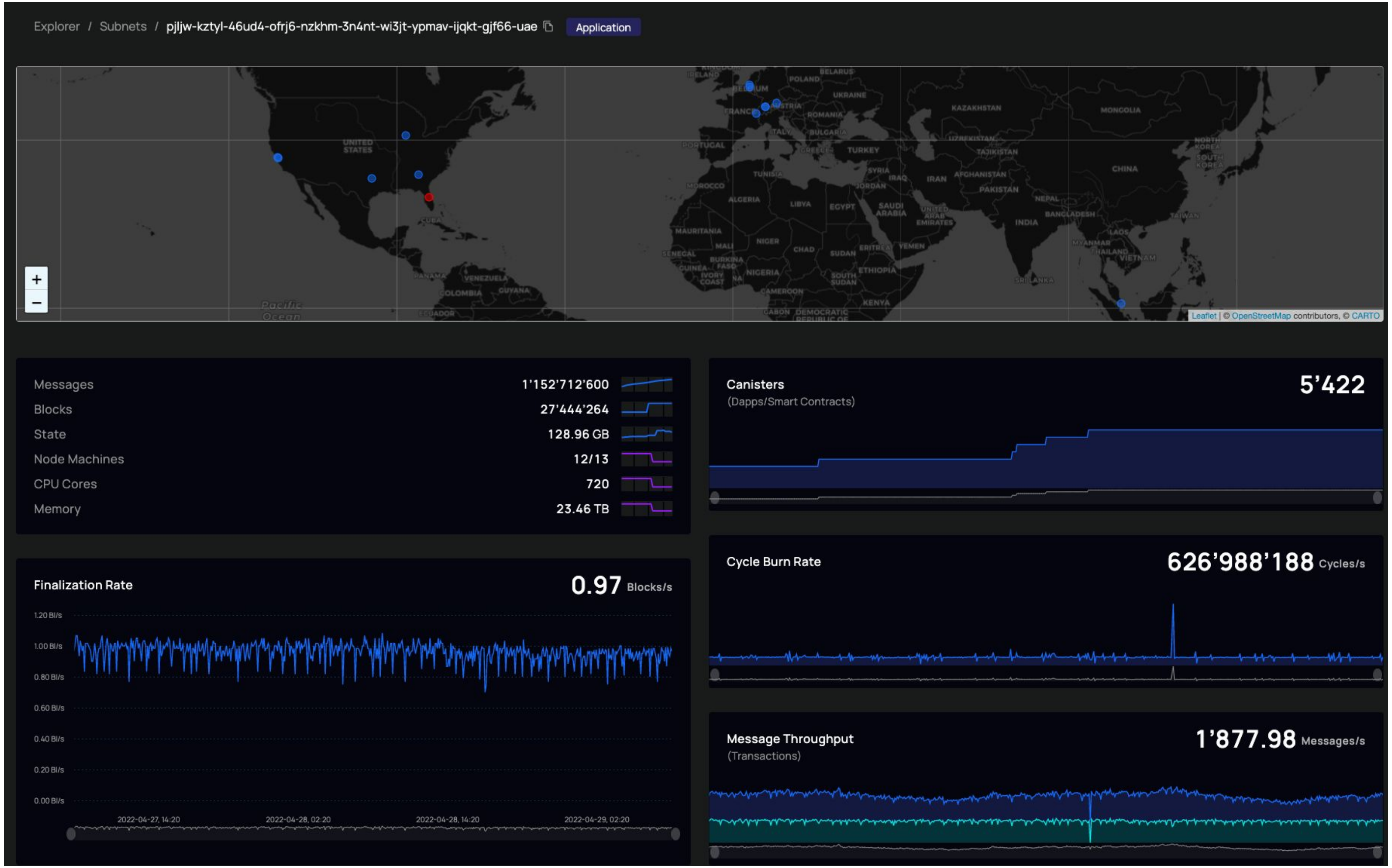
- Avg ⅔ dissemination latency:
  - NNS avg=1.39s, 95%=3.3s
  - App avg=0.57s, 95%=1.1s



https://dashboard.internetcomputer.org/

DFINITY

Example Subnet Dashboard



DFINITY

Research on the IC

Open Research Problems

- Intra-subnet communications scalability (growing size of subnets)
- Inter-subnet communications scalability (growing number of subnets)
- Ongoing firewall rule management
- Resilience against malicious activity
- Monitoring of node and network behavior
- Dynamic load balancing
- Caching
- Canister addressing

DFINITY

Testnets

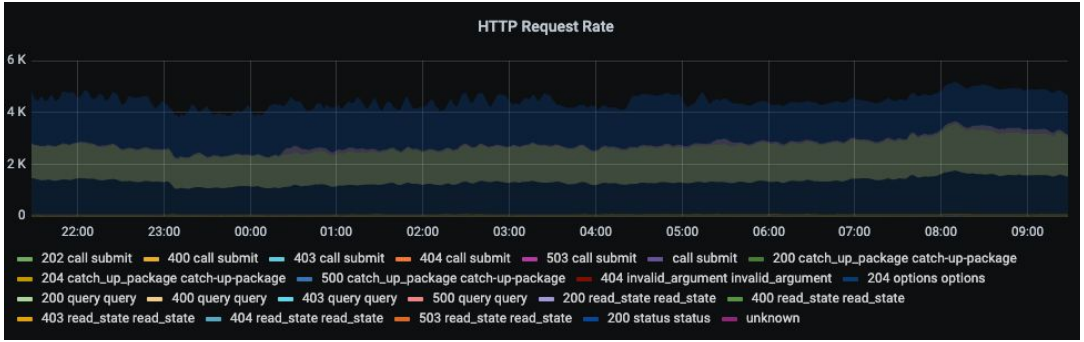
DFINITY-internal infrastructure

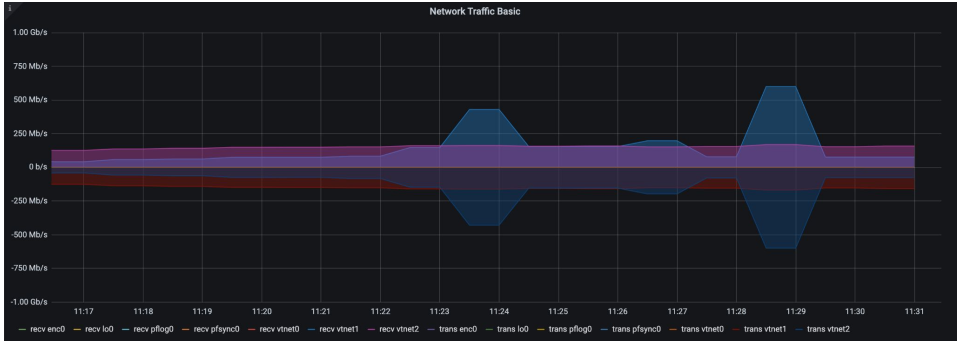
- Deploy complete IC instances in our 5 data centers (2 more in May)
    - Chicago, San Francisco, Des Moines, Frankfurt, Zurich, ..
- Variable size and VM capabilities
- Can be used for experiments, metrics, correctness and performance tests

Metrics

We can collect two general types of metrics:

- Code metrics
  - On mainnet
  - On testnets
- Infrastructure metrics





DFINITY

Logging

- Events can be logged in the code
- Log can be fetched from testnet machines
- Policy monitoring with MonPoly from Prof. Basin’s group

Case Study: “Idle” vs. Workload Traffic

31 nodes deployment

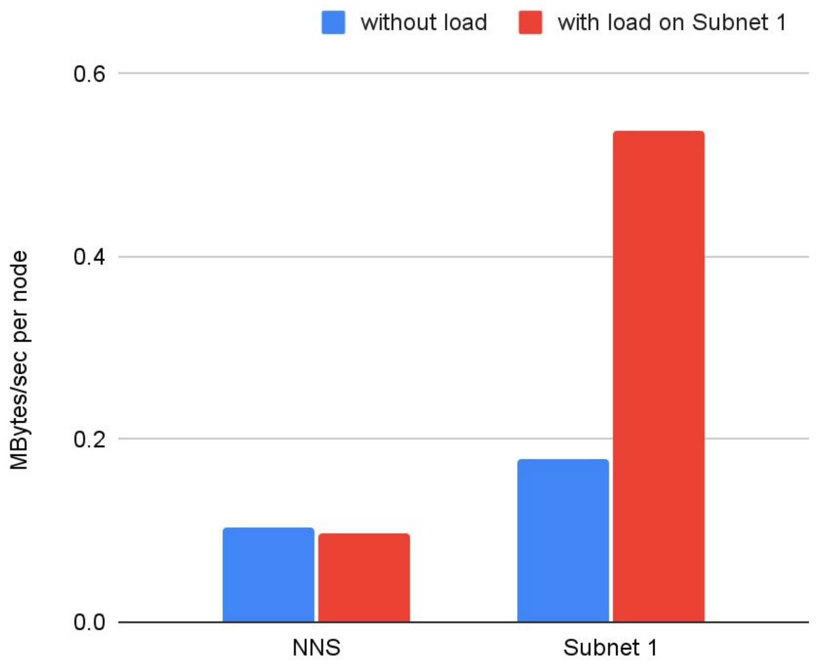
- 13 in NNS
- 18 in Subnet 1

Workload generation

- only in Subnet 1
- 100 requests per sec
- 1 kb each

Conclusion

- ICP produces 0.1-0.2MBytes/s for the protocol to make progress.



Case Study: “Intra DC” == Internet

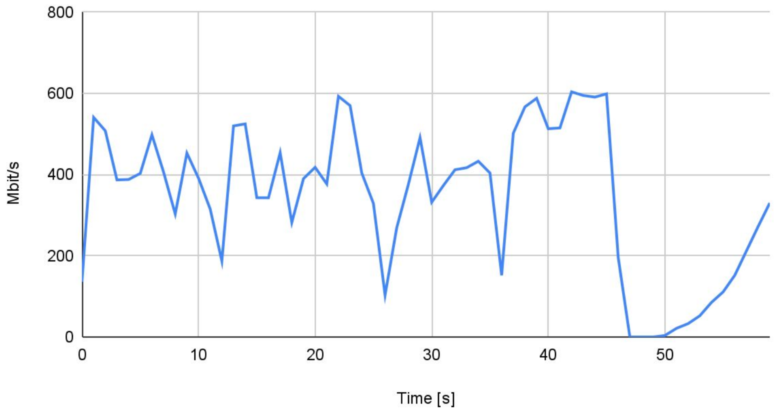
iperf between testnet hosts

- Chicago to San Francisco
- 60s in total

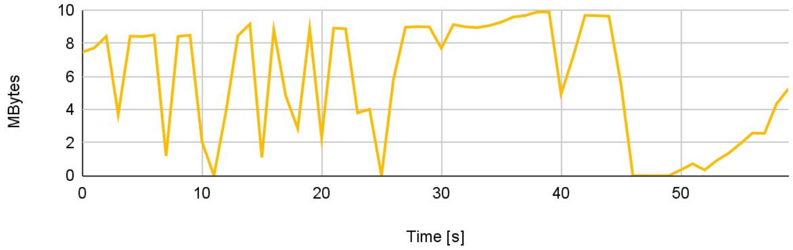
Conclusion

- Packet loss has a significant impact on the achieved throughput.

CH1 -> SF1 TCP (Throughput/Retransmissions)



CH1 -> SF1 TCP (CWND)



DFINITY

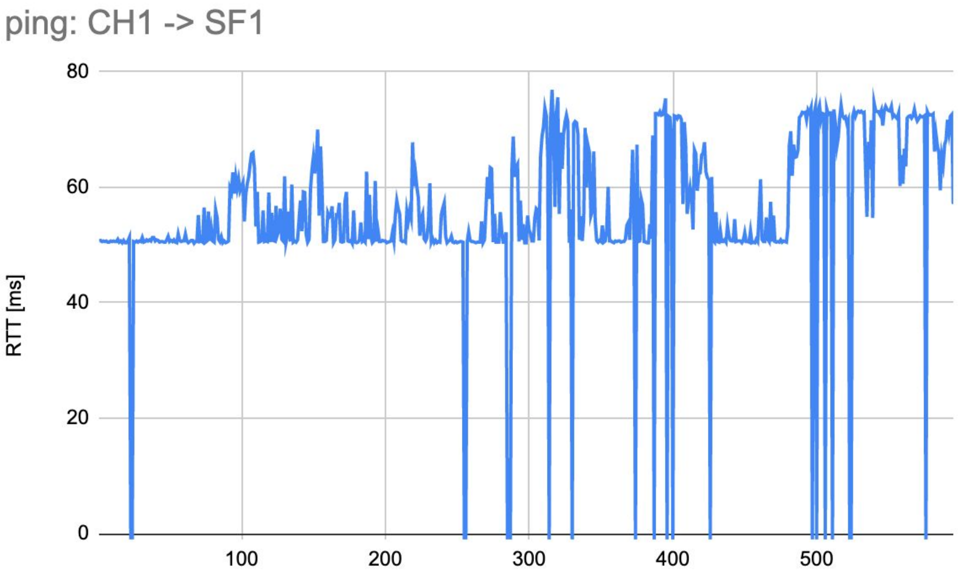
Case Study: RTT and Packet Loss

ping between testnet hosts

- Chicago to San Francisco
- every 1s for 10mins

Conclusion

- Extremely high packet loss (3.5%).
- Path change towards the end?



DFINITY

yotam.harchol@dfinity.org

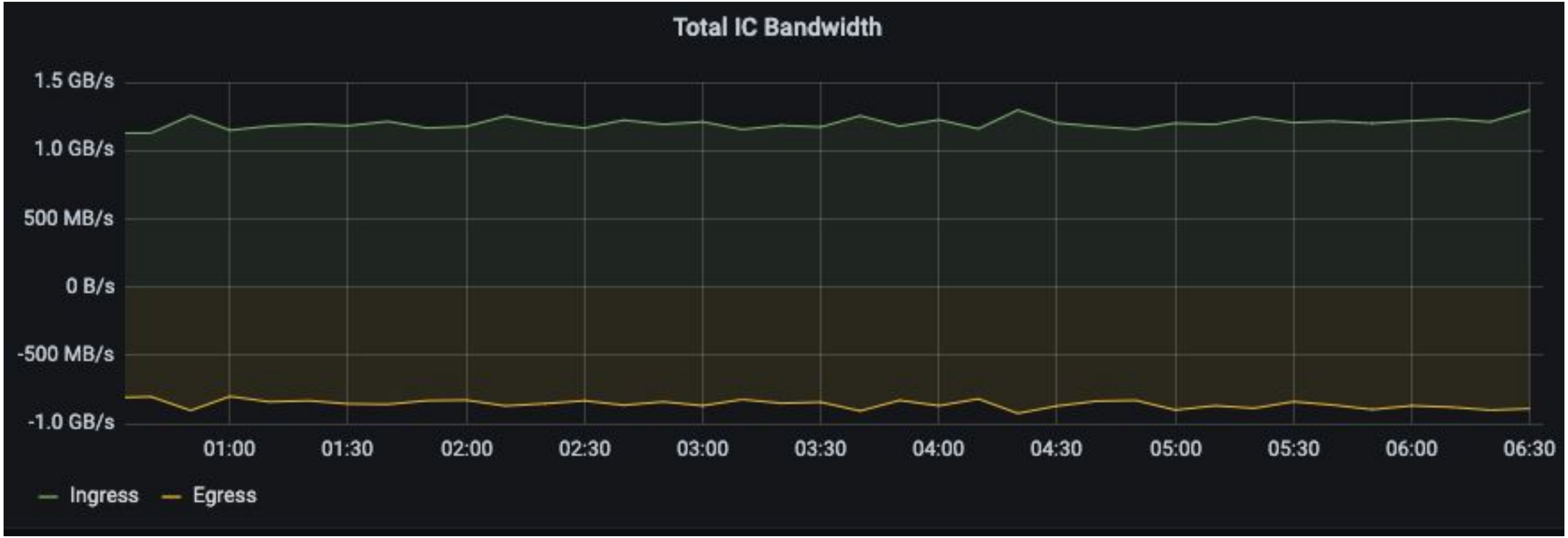
yvonneanne.pignolet@dfinity.org

ruediger.birkner@dfinity.org

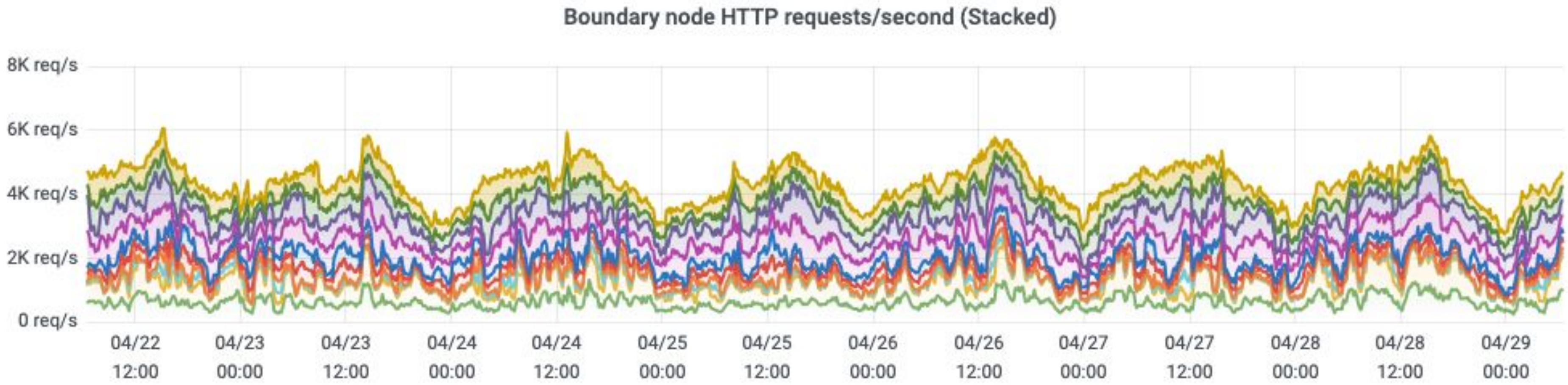
We are hiring: dfinity.org/careers

More information

- Infographic: here
- Technical Library: here (videos of talks) and here (blogposts)
- 200,000,000 CHF Developer Grant Program here
- DFINITY SDK: here


Total IC Bandwidth
1.5 GB/s
1.0 GB/s
500 MB/s
0 B/s
-500 MB/s
-1.0 GB/s
01:00
01:30
02:00
02:30
03:00
03:30
04:00
04:30
05:00
05:30
06:00
06:30
Ingress
Egress

DFINITY


Boundary node HTTP requests/second (Stacked)
8K req/s
6K req/s
4K req/s
2K req/s
0 req/s
04/22 12:00
04/23 00:00
04/23 12:00
04/24 00:00
04/24 12:00
04/25 00:00
04/25 12:00
04/26 00:00
04/26 12:00
04/27 00:00
04/27 12:00
04/28 00:00
04/28 12:00
04/29 00:00

DFINITY