Node to node communication in Vantage6

Renee Witsenburg

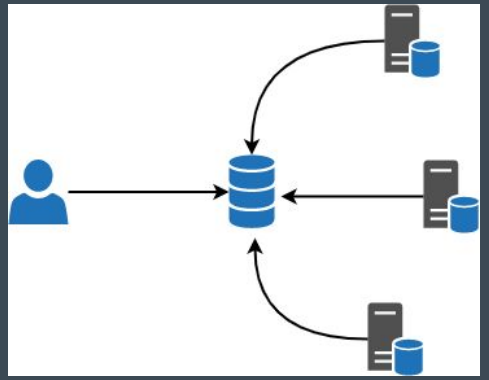
Traditional central approach

A data scientist has (a) research question(s) that needs data from multiple data sources

Copying the datasets and centralizing them

Disadvantages:

- Transport
- Loss of control
- Privacy
- Security
- Law



Personal Health Train

- Patient data (possibly enriched with personal health data) stored in data stations
- Data stations can be clustered together in larger data stations
- Algorithms/trains are moved to the data stations by researchers

- Data owners have access to their own data and can set access permissions

"Coronary artery disease: risk estimations and interventions for prevention and Early detection – a personal health train project"

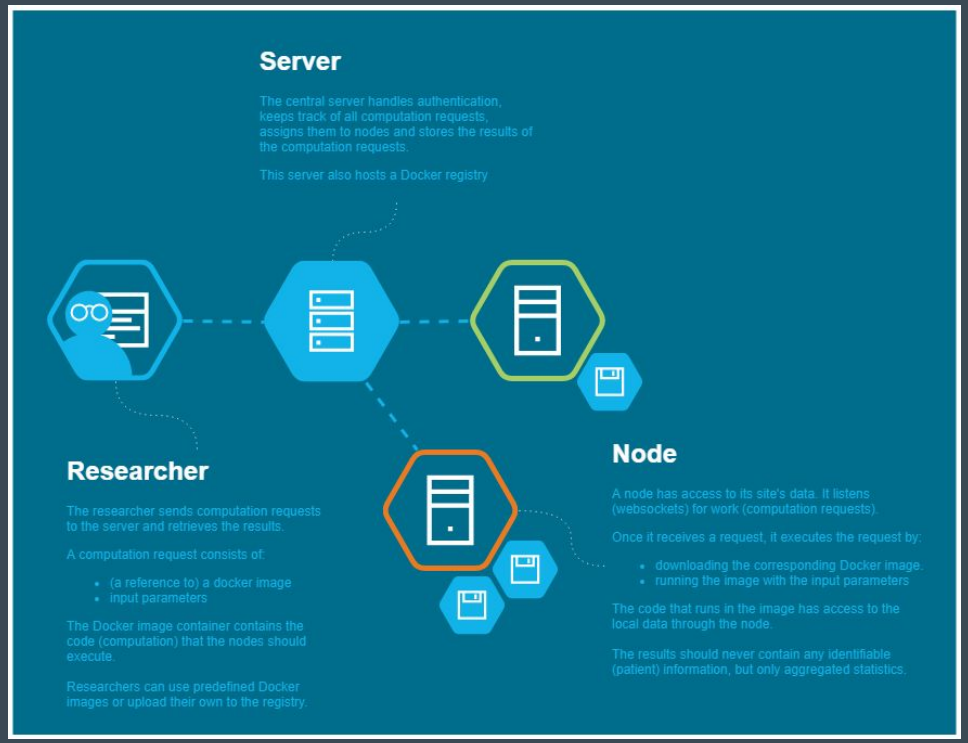
- Carrier project

netherlands eScience center Maastricht Univ

Vantage6

Infrastructure to do federated learning

- Vantage6 server
- Vantage6 nodes
- Collaboration
- Organisation
- Registries



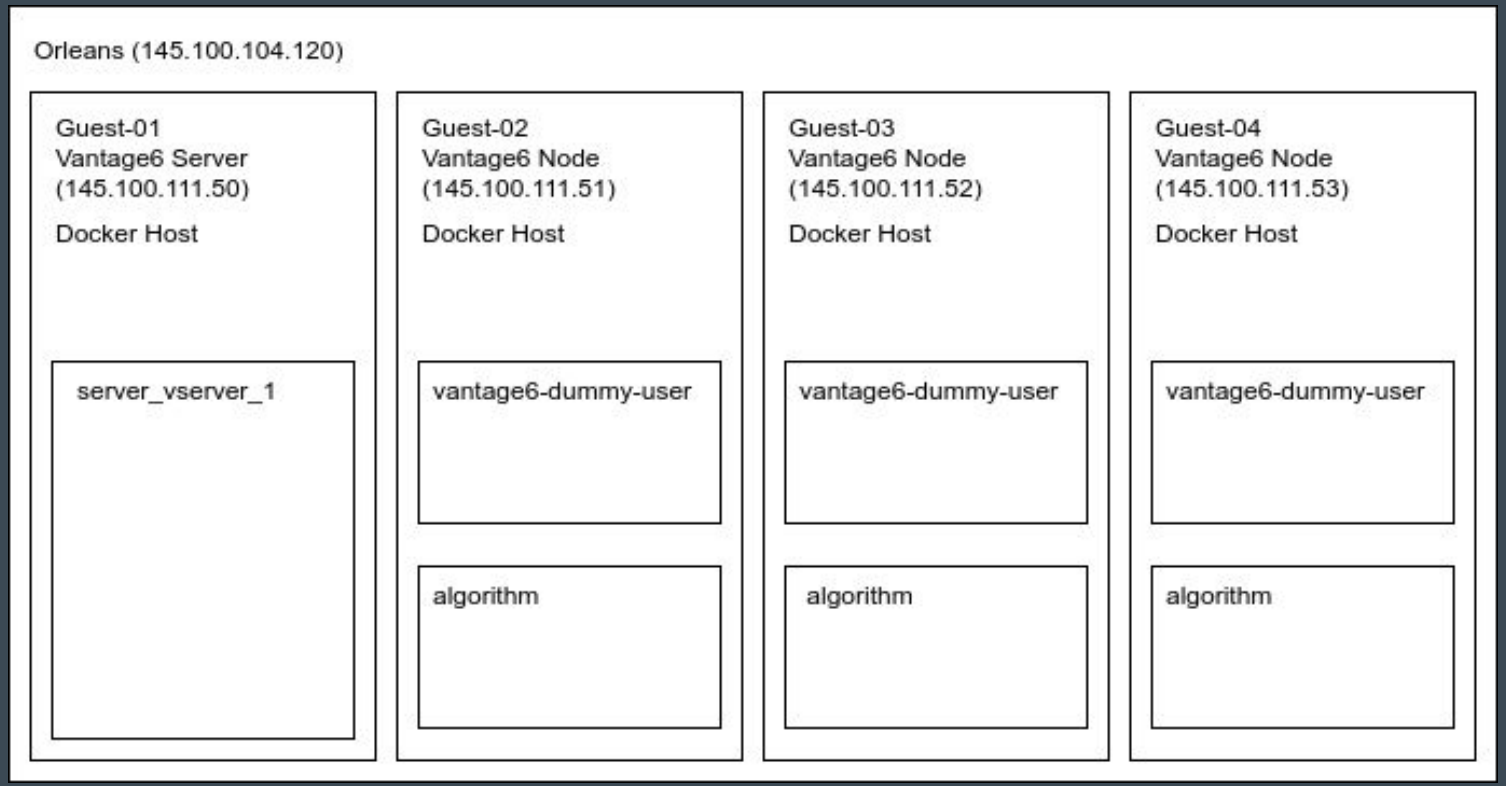
Source: https://vantage6.ai/blog-index/about-secure-multi-party-computation/

Research questions

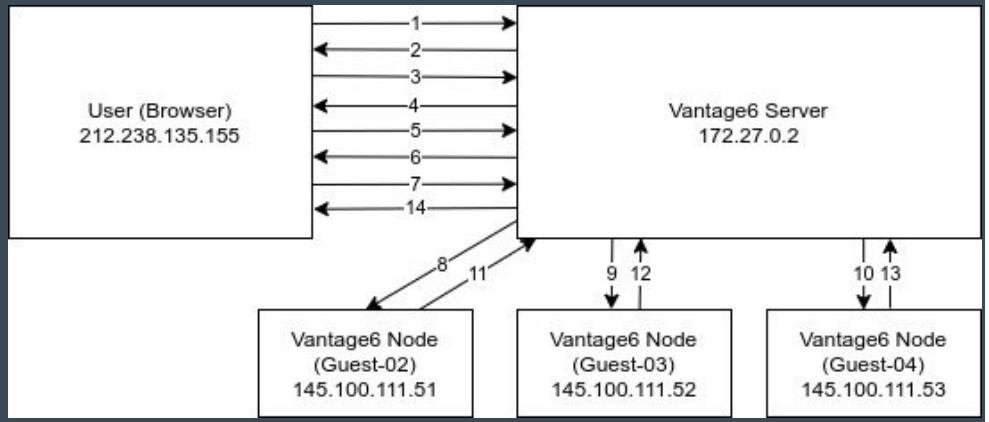
How can Vantage6 edge nodes work together efficiently, without the interposition of the central server?

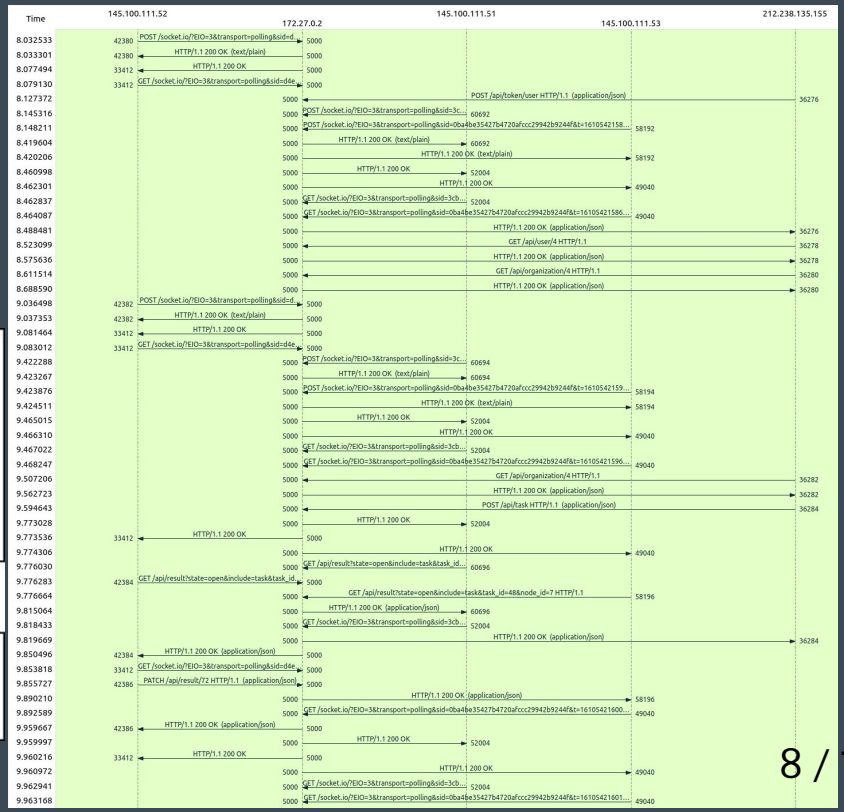
- What are the issues in the current Vantage6 project that create a bottleneck between working nodes?
- Which infrastructures could be implemented to make the nodes work together?
- Is it possible to tamper with, read or intercept data or the model?
- How are malicious attempts to corrupt the data or the model detected?

Test setup Vantage6



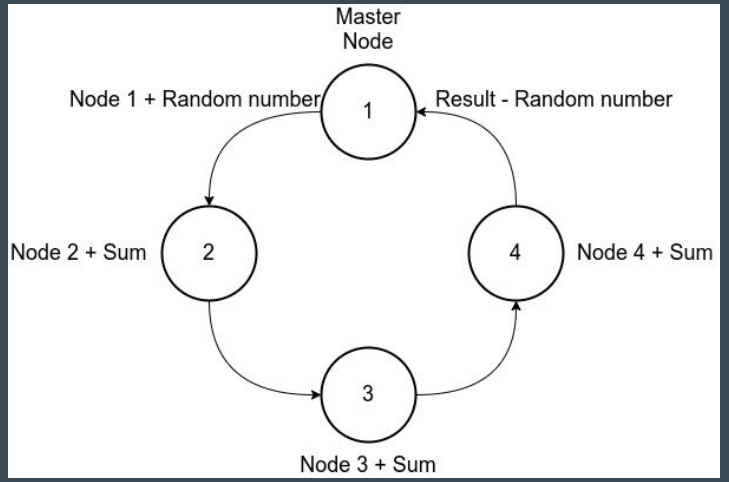
Network traffic inside the Vantage6 infrastructure



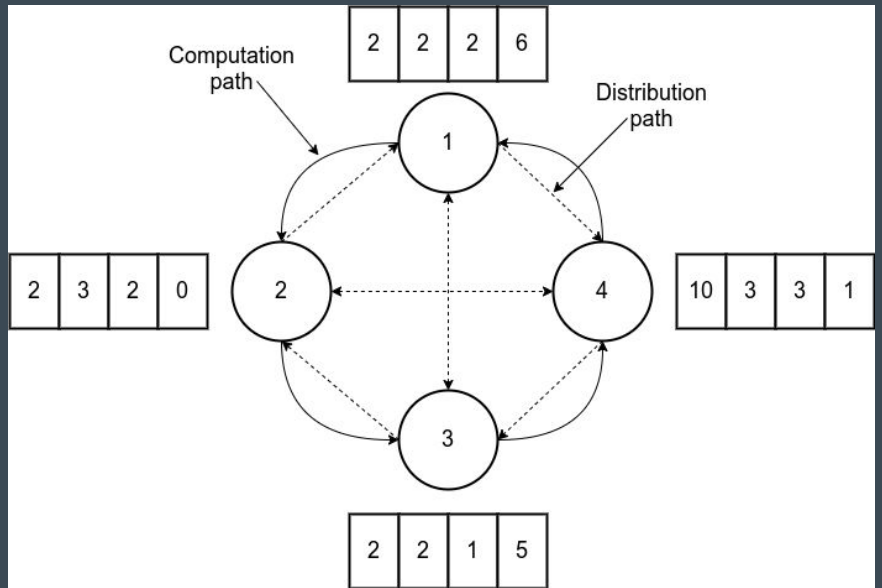


Nodes working together

Secure sum

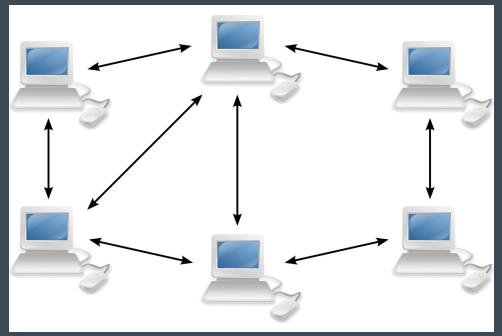


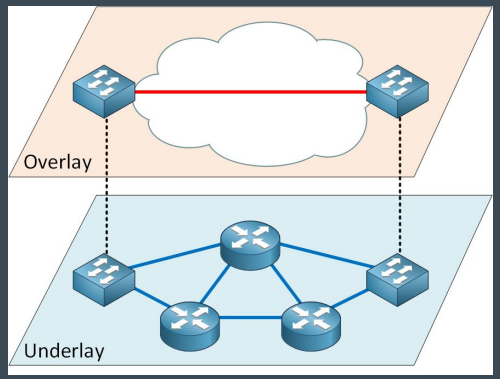
dk-Secure sum



Node to node communication

- Node to node communication with the Vantage6 server in the middle
- Node to node communication without the Vantage6 server in the middle
  - Peer-to-peer
  - VPN
  - Overlay





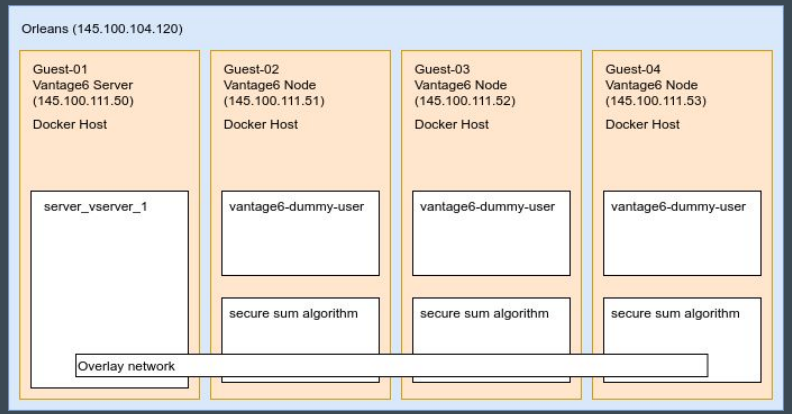
Proof of Concept (Overlay)

Docker Swarm: Vantage6 server is docker swarm manager

Create overlay network on the Vantage6 server and share the swarm IP + token with the Vantage6 nodes in the collaboration

Vantage6 nodes in de collaboration connect to the swarm

Algorithm containers connect to the overlay network



Proof of concept

vxlan

No.	Time	Source	Destination	Protocol	Length	Info
→ 77	3.053460	10.0.1.67	10.0.1.65	ICMP	148	Echo (ping) request id=0x0f00, seq=0/0, ttl=64 (reply in 78)
← 78	3.053506	10.0.1.65	10.0.1.67	ICMP	148	Echo (ping) reply id=0x0f00, seq=0/0, ttl=64 (request in 77)
103	4.053578	10.0.1.67	10.0.1.65	ICMP	148	Echo (ping) request id=0x0f00, seq=1/256, ttl=64 (reply in 104)
104	4.053653	10.0.1.65	10.0.1.67	ICMP	148	Echo (ping) reply id=0x0f00, seq=1/256, ttl=64 (request in 103)
115	5.053735	10.0.1.67	10.0.1.65	ICMP	148	Echo (ping) request id=0x0f00, seq=2/512, ttl=64 (reply in 116)
116	5.053796	10.0.1.65	10.0.1.67	ICMP	148	Echo (ping) reply id=0x0f00, seq=2/512, ttl=64 (request in 115)
122	6.053830	10.0.1.67	10.0.1.65	ICMP	148	Echo (ping) request id=0x0f00, seq=3/768, ttl=64 (reply in 123)
123	6.053863	10.0.1.65	10.0.1.67	ICMP	148	Echo (ping) reply id=0x0f00, seq=3/768, ttl=64 (request in 122)
138	7.054000	10.0.1.67	10.0.1.65	ICMP	148	Echo (ping) request id=0x0f00, seq=4/1024, ttl=64 (reply in 139)
139	7.054060	10.0.1.65	10.0.1.67	ICMP	148	Echo (ping) reply id=0x0f00, seq=4/1024, ttl=64 (request in 138)
148	8.054072	10.0.1.67	10.0.1.65	ICMP	148	Echo (ping) request id=0x0f00, seq=5/1280, ttl=64 (reply in 149)
149	8.054121	10.0.1.65	10.0.1.67	ICMP	148	Echo (ping) reply id=0x0f00, seq=5/1280, ttl=64 (request in 148)
182	9.054240	10.0.1.67	10.0.1.65	ICMP	148	Echo (ping) request id=0x0f00, seq=6/1536, ttl=64 (reply in 183)
183	9.054293	10.0.1.65	10.0.1.67	ICMP	148	Echo (ping) reply id=0x0f00, seq=6/1536, ttl=64 (request in 182)
194	10.054366	10.0.1.67	10.0.1.65	ICMP	148	Echo (ping) request id=0x0f00, seq=7/1792, ttl=64 (reply in 195)
195	10.054430	10.0.1.65	10.0.1.67	ICMP	148	Echo (ping) reply id=0x0f00, seq=7/1792, ttl=64 (request in 194)
202	11.054517	10.0.1.67	10.0.1.65	ICMP	148	Echo (ping) request id=0x0f00, seq=8/2048, ttl=64 (reply in 203)
203	11.054577	10.0.1.65	10.0.1.67	ICMP	148	Echo (ping) reply id=0x0f00, seq=8/2048, ttl=64 (request in 202)
237	12.054588	10.0.1.67	10.0.1.65	ICMP	148	Echo (ping) request id=0x0f00, seq=9/2304, ttl=64 (reply in 238)
238	12.054631	10.0.1.65	10.0.1.67	ICMP	148	Echo (ping) reply id=0x0f00, seq=9/2304, ttl=64 (request in 237)

```
▸ Frame 77: 148 bytes on wire (1184 bits), 148 bytes captured (1184 bits)
▸ Ethernet II, Src: Xensourc_a4:29:e1 (00:16:3e:a4:29:e1), Dst: Xensourc_50:2d:65 (00:16:3e:50:2d:65)
▾ Internet Protocol Version 4, Src: 145.100.111.52, Dst: 145.100.111.51
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 134
    Identification: 0x5390 (21392)
  ▸ Flags: 0x0000
    Fragment offset: 0
    Time to live: 64
    Protocol: UDP (17)
    Header checksum: 0x25a7 [validation disabled]
    [Header checksum status: Unverified]
    Source: 145.100.111.52
    Destination: 145.100.111.51
▸ User Datagram Protocol, Src Port: 38950, Dst Port: 4789
▾ Virtual eXtensible Local Area Network
  ▸ Flags: 0x0800, VXLAN Network ID (VNI)
    Group Policy ID: 0
    VXLAN Network Identifier (VNI): 4097
    Reserved: 0
▾ Ethernet II, Src: 02:42:0a:00:01:43 (02:42:0a:00:01:43), Dst: 02:42:0a:00:01:41 (02:42:0a:00:01:41)
  ▸ Destination: 02:42:0a:00:01:41 (02:42:0a:00:01:41)
  ▸ Source: 02:42:0a:00:01:43 (02:42:0a:00:01:43)
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 10.0.1.67, Dst: 10.0.1.65
▸ Internet Control Message Protocol
```

Proof of concept

“All swarm service management traffic is encrypted by default, using the AES algorithm in GCM mode. Manager nodes in the swarm rotate the key used to encrypt gossip data every 12 hours.”

“To encrypt application data as well, add `--opt encrypted` when creating the overlay network. This enables IPSEC encryption at the level of the vxlan.”

With overlay encryption enabled, Docker creates IPSEC tunnels between nodes. Keys rotate every 12 hours.

Discussion

Only researched overlay networks

Main focus on node to node communication and not security

All nodes in the infrastructure need to be in the swarm to connect to the overlay network

Setup can be recreated without a swarm but with a key/value container

Conclusion

Nodes can interact with each other through an overlay network

Vantage6 still has a lot of bugs and missing features

Future work

Other federated learning frameworks like PyGrid an PySyft

Other node to node communication performance

Developing Vantage6 and Docker-py further

Questions?