KubeCon | CloudNativeCon

North America 2023

Making Kubernetes Quantum-Safe: What can we do to protect ourselves now?

Paul Schweigert and Michael (Max)imilien, IBM

Speakers

Paul Schweigert (pssschwei.com)

Senior Software Engineer at IBM

Knative Technical Oversight Committee

Qiskit Advocate

Dr. Max (@maximilien)

Distinguished Engineer at IBM

CTO Open Quantum and Open Serverless

Cyclist / photographer

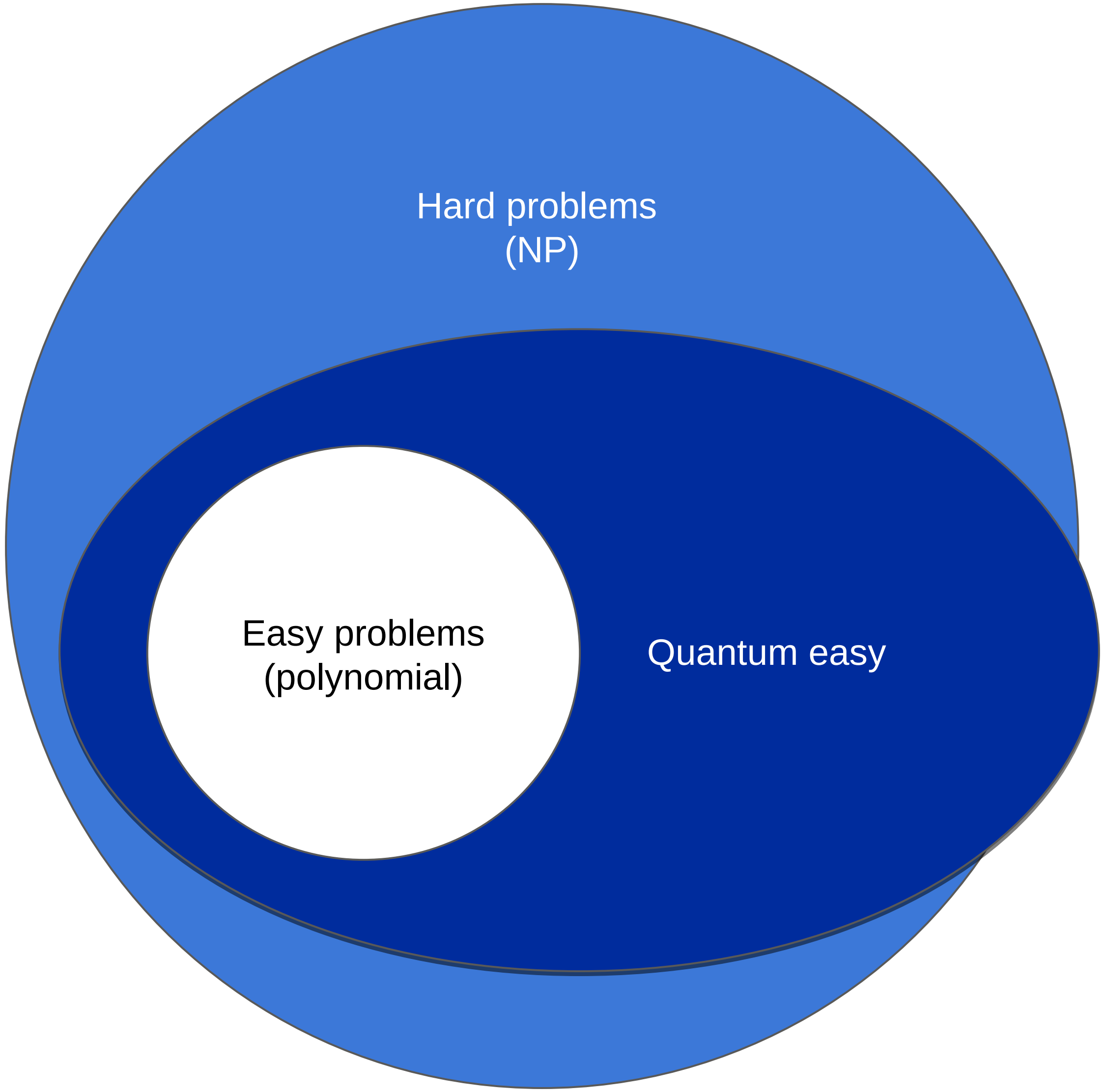
Agenda

- What is quantum computing?
- What is the quantum threat?
- How do we protect Kubernetes?

Agenda

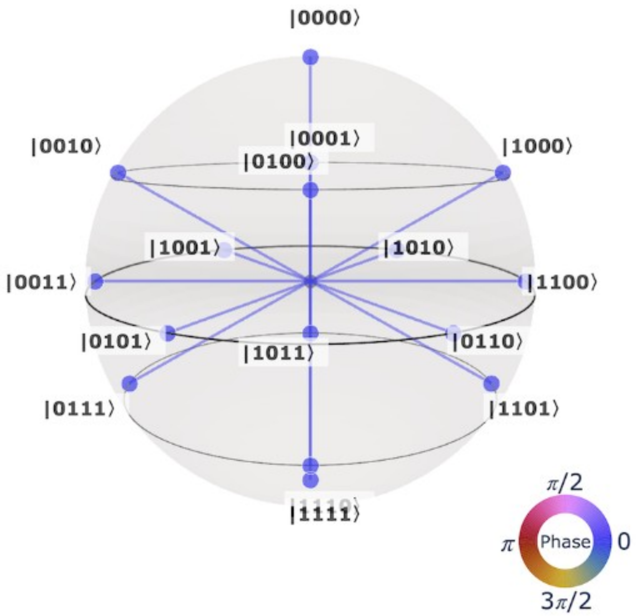
- What is quantum computing?
- What is the quantum threat?
- How do we protect Kubernetes?

Why quantum?

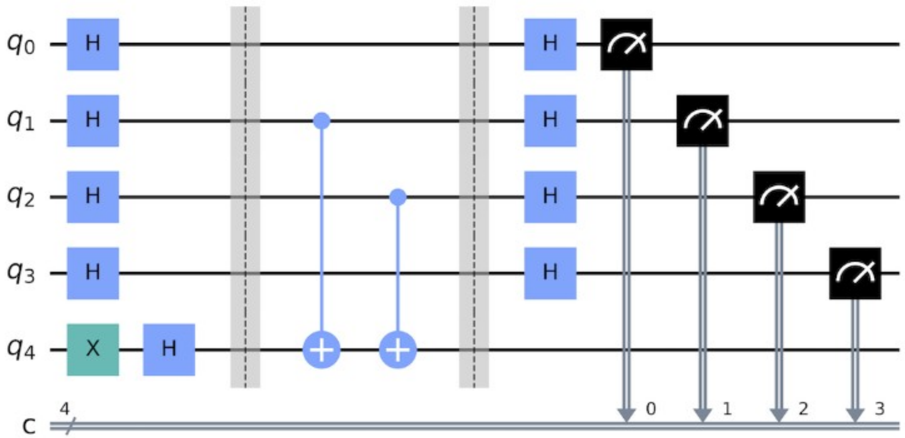


Quantum computers use qubits

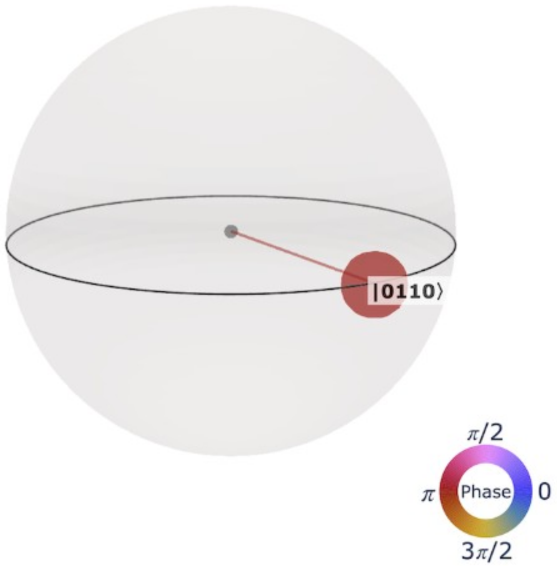
Quantum circuit



Superposition of
all possibilities

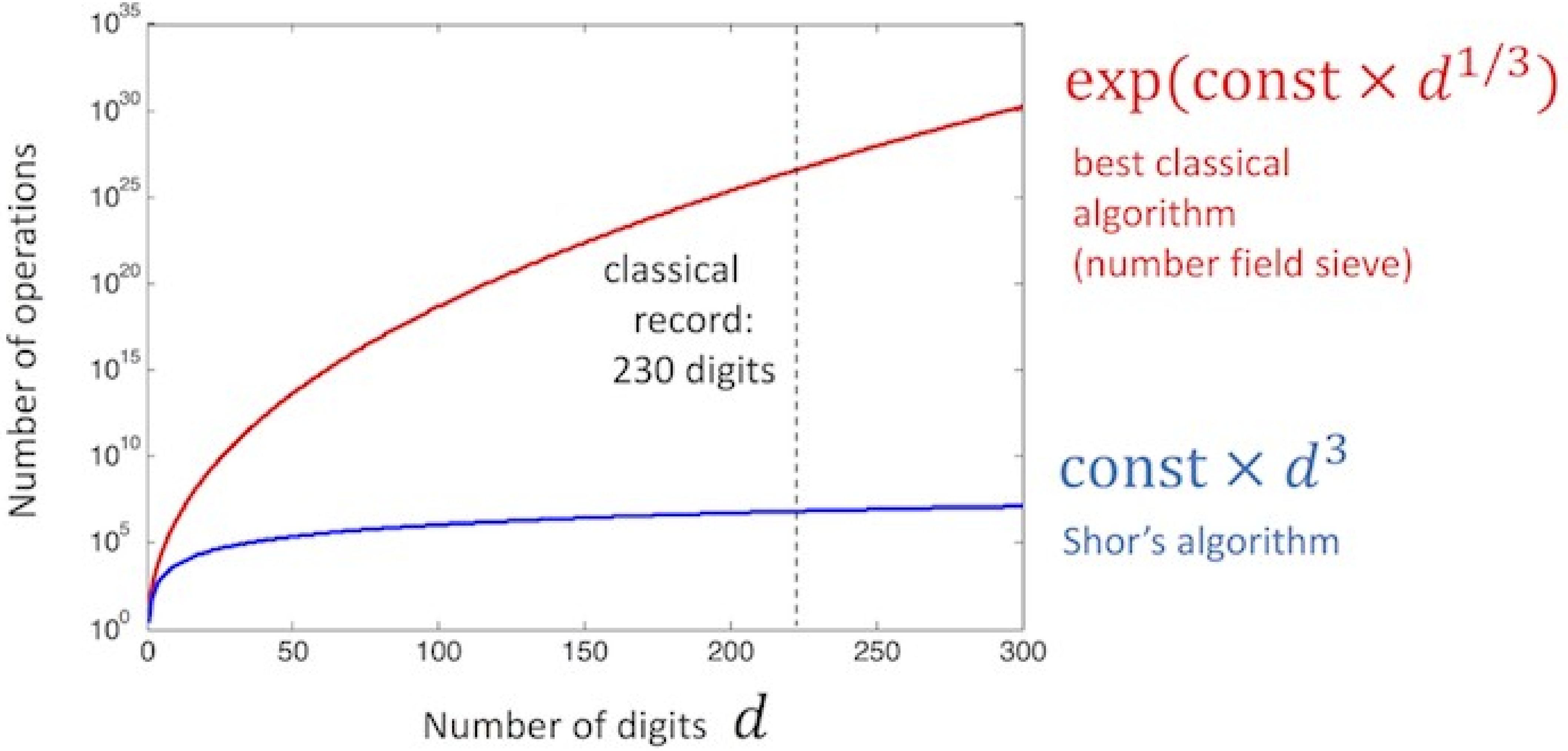


Computation driven interference



Solution

Ex: Shor’s algorithm for factoring



Agenda

- What is quantum computing?
- What is the quantum threat?
- How do we protect Kubernetes?

Current cryptography is at risk

Prime factors

$= p \times q$

2048-bit composite integer

```
25195908475657893494027183240048398571429282126204032027
77771378360436620207075955562640185258807844069182906412
49515082189298559149176184502808489120072844992687392807
28777673597141834727026189637501497182469116507761337
98590957000973304597488084284017974291006424586918171951
18746121515172654632282216869987549182422433637259085141
86546204357679842338718477444792073993423658482382428
11981638150106748104516603773060562016196762561338441436
03833904414952634432190114657544454178424020924616515723
35077870774981712577246796292638635637328991215483143
81678998850404453640235273819513786365643912120103971228
22120720357
```

Expected computation time

The most powerful computer **today:**

Millions of years

Shor's quantum algorithm:

Hours

Per Shor's algorithm, all public key crypto standards are vulnerable to attacks from large scale quantum computers

Public Key Encryption	RSA
Digital Signatures	DSA, ECDSA
Key Exchange Algorithms	Diffie-Hellman, ECDH

Our modern digital world depends on cryptography

It is the ultimate line of defense

Trillions of Transactions on Billions of Devices use cryptography - including cellphones, laptops, desktops, services, ATMs, Internet Routers, VPN Servers, Smart IoT

What will a cybercriminal be able to do?

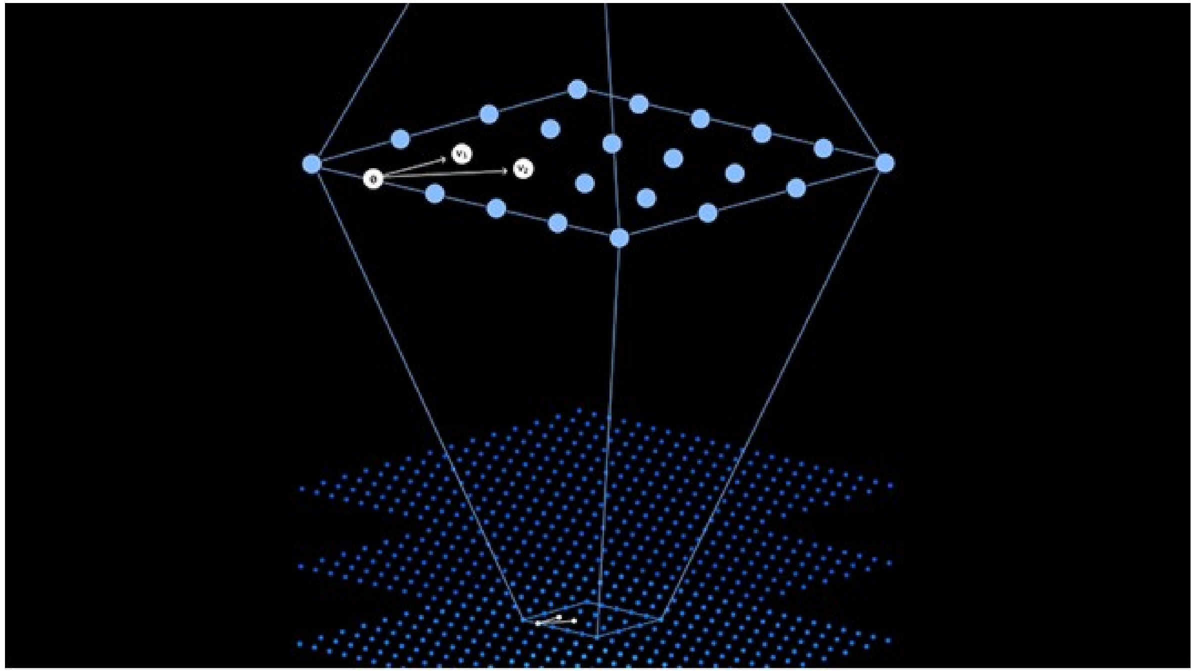
- Fraudulent authentication
- Forge digital signatures
- Harvest now, decrypt later

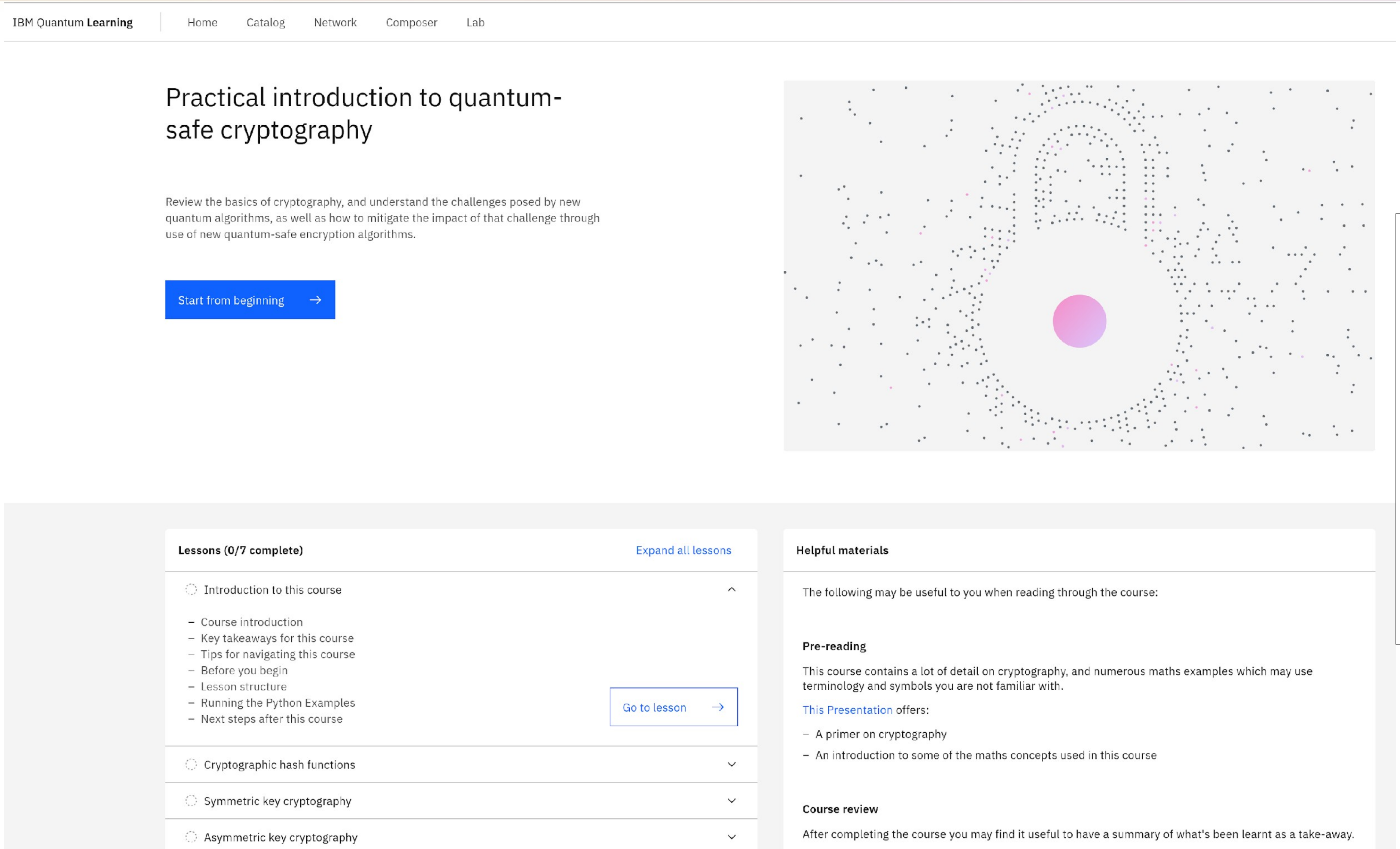
Quantum Safe Cryptography

a.k.a. Post Quantum Cryptography or Quantum Resistant Cryptography

Traditional public-key cryptography relies upon mathematical problems that are difficult to solve on classical computers.

Quantum-safe cryptography includes a suite of algorithms and systems that are resistant to attacks by both classical and quantum computers.





Practical introduction to quantum-safe cryptography

Review the basics of cryptography, and understand the challenges posed by new quantum algorithms, as well as how to mitigate the impact of that challenge through use of new quantum-safe encryption algorithms.

Lessons (0/7 complete)

Introduction to this course

- Course introduction
- Key takeaways for this course
- Tips for navigating this course
- Before you begin
- Lesson structure
- Running the Python Examples
- Next steps after this course

Cryptographic hash functions

Symmetric key cryptography

Asymmetric key cryptography

Helpful materials

The following may be useful to you when reading through the course:

Pre-reading

This course contains a lot of detail on cryptography, and numerous maths examples which may use terminology and symbols you are not familiar with.

This Presentation offers:

- A primer on cryptography
- An introduction to some of the maths concepts used in this course

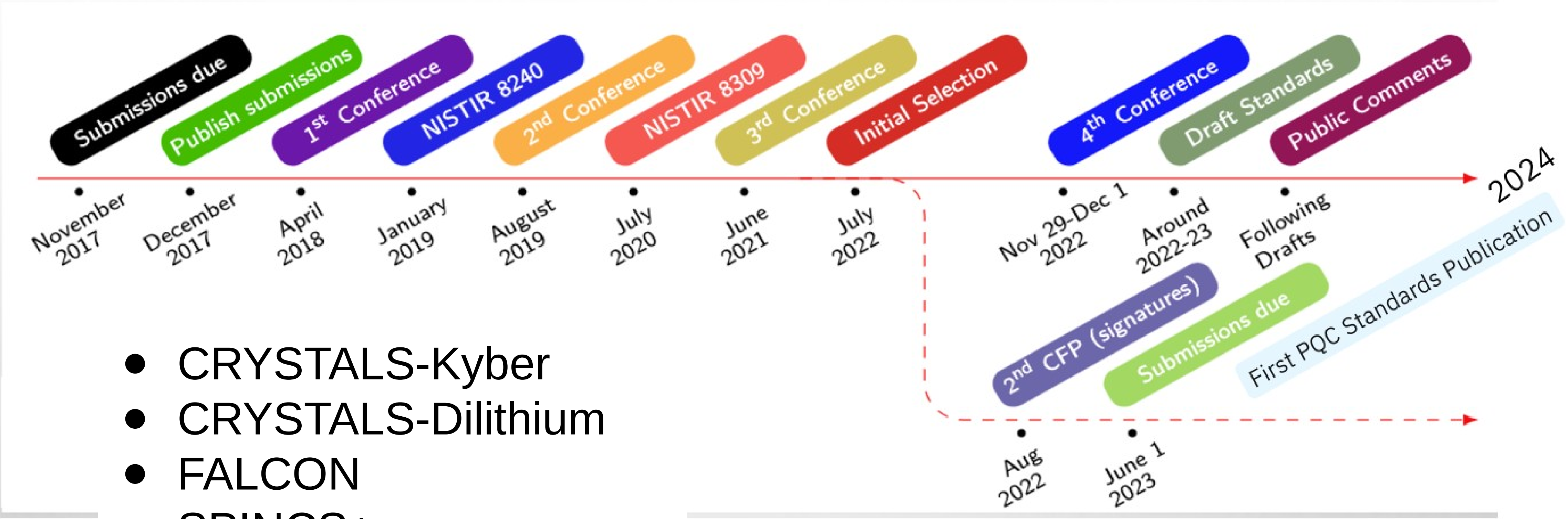
Course review

After completing the course you may find it useful to have a summary of what's been learnt as a take-away.

https://learning.quantum-computing.ibm.com/course/practical-introduction-to-quantum-safe-cryptography

National Institute of Standards and Technology (NIST)

Post Quantum Cryptography (PQC) Standardization Progress



Submissions due – November 2017

Publish submissions – December 2017

1st Conference – April 2018

NISTIR 8240 – January 2019

2nd Conference – August 2019

NISTIR 8309 – July 2020

3rd Conference – June 2021

Initial Selection – July 2022

4th Conference – Nov 29-Dec 1 2022

Draft Standards – Around 2022-23

Public Comments – Following Drafts

2024

2nd CFP (signatures) – Aug 2022

Submissions due – June 1 2023

First PQC Standards Publication

- CRYSTALS-Kyber
- CRYSTALS-Dilithium
- FALCON
- SPINCS+

Source: NIST

Open Quantum Safe

https://openquantumsafe.org/

Core team from: University of Waterloo, IBM, AWS, Microsoft, baentsch.ch

liboqs: an open source C library for quantum-safe cryptographic algorithms

It provides:

- a collection of open-source implementations of quantum-safe algorithms
- a common API
- tests and benchmarks

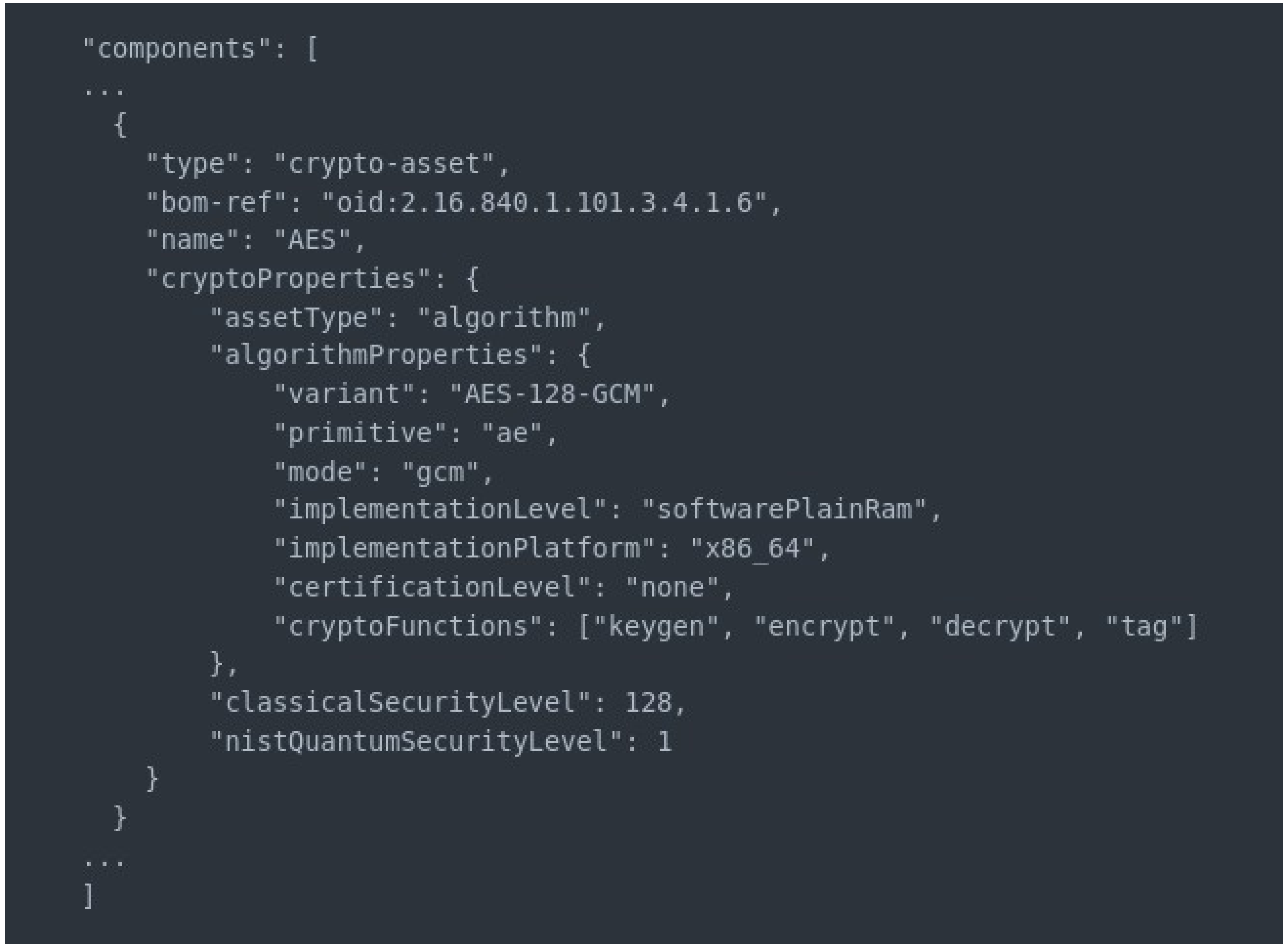
Applications:

- TLS
- SSH
- X.509
- CMS and S/MIME

Cryptography Bill of Materials (CBOM)

CBOM: an object model to describe crypto-assets and their dependencies

- Model crypto assets
- Capture crypto asset properties
- Capture crypto asset dependencies
- Applicable to various software components
- High compatibility with SBOMs and relating

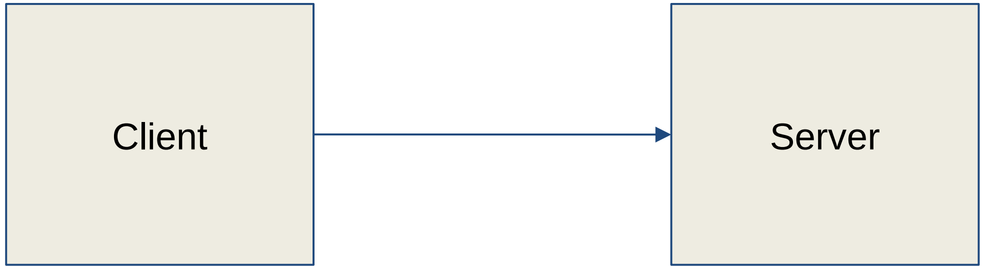


```
"components": [
...
  {
    "type": "crypto-asset",
    "bom-ref": "oid:2.16.840.1.101.3.4.1.6",
    "name": "AES",
    "cryptoProperties": {
        "assetType": "algorithm",
        "algorithmProperties": {
            "variant": "AES-128-GCM",
            "primitive": "ae",
            "mode": "gcm",
            "implementationLevel": "softwarePlainRam",
            "implementationPlatform": "x86_64",
            "certificationLevel": "none",
            "cryptoFunctions": ["keygen", "encrypt", "decrypt", "tag"]
        },
        "classicalSecurityLevel": 128,
        "nistQuantumSecurityLevel": 1
    }
  }
...
]
```

Agenda

- What is quantum computing?
- What is the quantum threat?
- How do we protect Kubernetes?

Where do we need to be secure?



Demo: Quantum-Safe connection

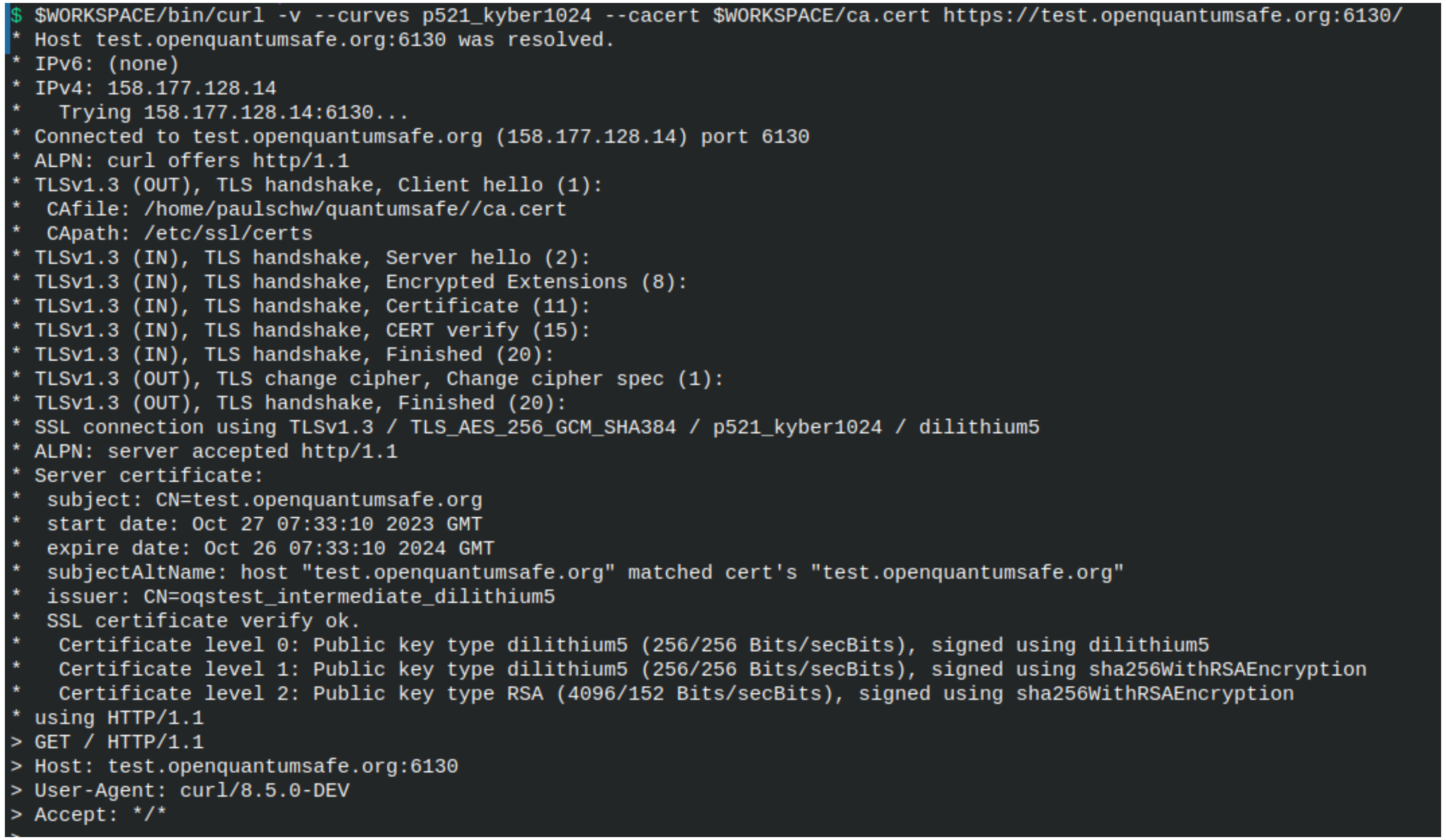
1. Build OpenSSL

2. Build liboqs

3. Install Open Quantum Safe provider

4. Build curl with OQS provider

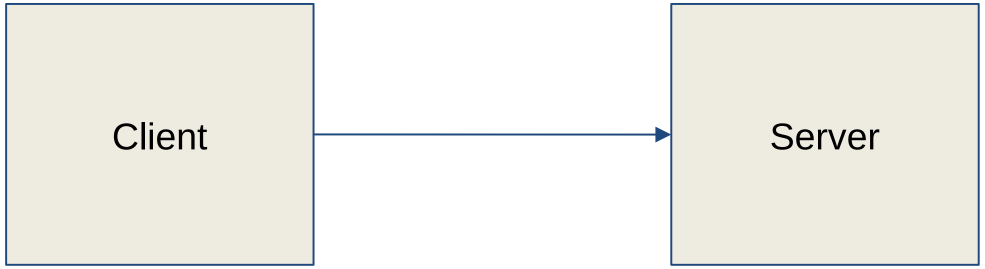
5. Connect!



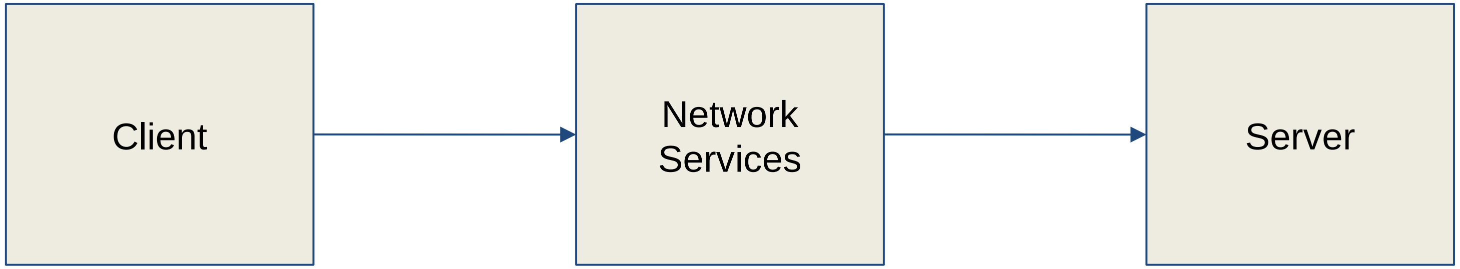
```
$ $WORKSPACE/bin/curl -v --curves p521_kyber1024 --cacert $WORKSPACE/ca.cert https://test.openquantumsafe.org:6130/
* Host test.openquantumsafe.org:6130 was resolved.
* IPv6: (none)
* IPv4: 158.177.128.14
*   Trying 158.177.128.14:6130...
* Connected to test.openquantumsafe.org (158.177.128.14) port 6130
* ALPN: curl offers http/1.1
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
*  CAfile: /home/paulschw/quantumsafe//ca.cert
*  CApath: /etc/ssl/certs
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.3 (IN), TLS handshake, Encrypted Extensions (8):
* TLSv1.3 (IN), TLS handshake, Certificate (11):
* TLSv1.3 (IN), TLS handshake, CERT verify (15):
* TLSv1.3 (IN), TLS handshake, Finished (20):
* TLSv1.3 (OUT), TLS change cipher, Change cipher spec (1):
* TLSv1.3 (OUT), TLS handshake, Finished (20):
* SSL connection using TLSv1.3 / TLS_AES_256_GCM_SHA384 / p521_kyber1024 / dilithium5
* ALPN: server accepted http/1.1
* Server certificate:
*  subject: CN=test.openquantumsafe.org
*  start date: Oct 27 07:33:10 2023 GMT
*  expire date: Oct 26 07:33:10 2024 GMT
*  subjectAltName: host "test.openquantumsafe.org" matched cert's "test.openquantumsafe.org"
*  issuer: CN=oqstest_intermediate_dilithium5
*  SSL certificate verify ok.
*   Certificate level 0: Public key type dilithium5 (256/256 Bits/secBits), signed using dilithium5
*   Certificate level 1: Public key type dilithium5 (256/256 Bits/secBits), signed using sha256WithRSAEncryption
*   Certificate level 2: Public key type RSA (4096/152 Bits/secBits), signed using sha256WithRSAEncryption
* using HTTP/1.1
> GET / HTTP/1.1
> Host: test.openquantumsafe.org:6130
> User-Agent: curl/8.5.0-DEV
> Accept: */*
```

https://developer.ibm.com/tutorials/awb-quantum-safe-openssl/

Where do we need to be secure?

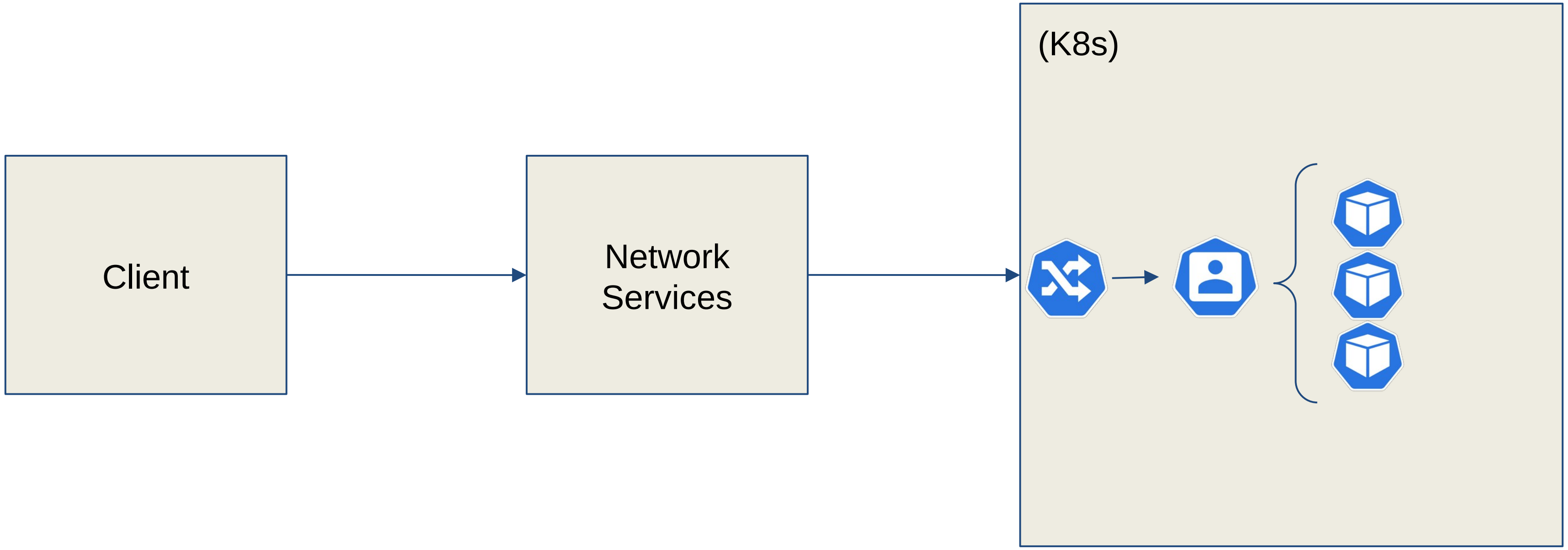


Where do we need to be secure?



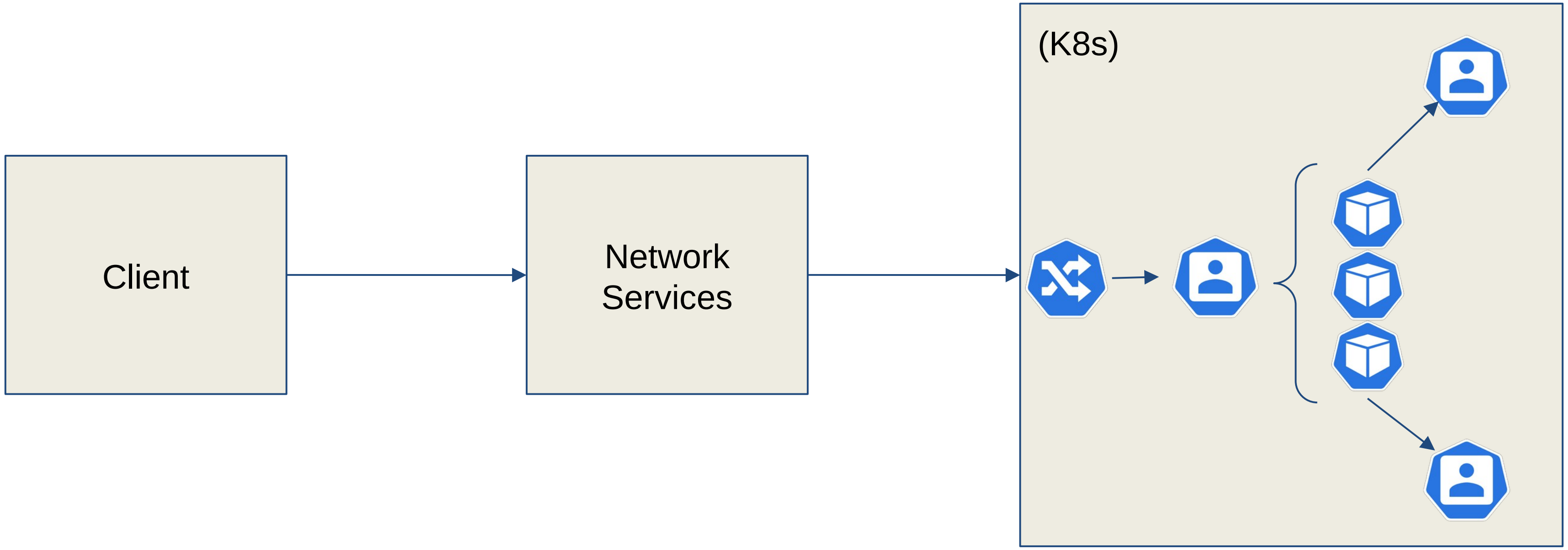
Where do we need to be secure?

Ingress -> Service -> Pod(s)



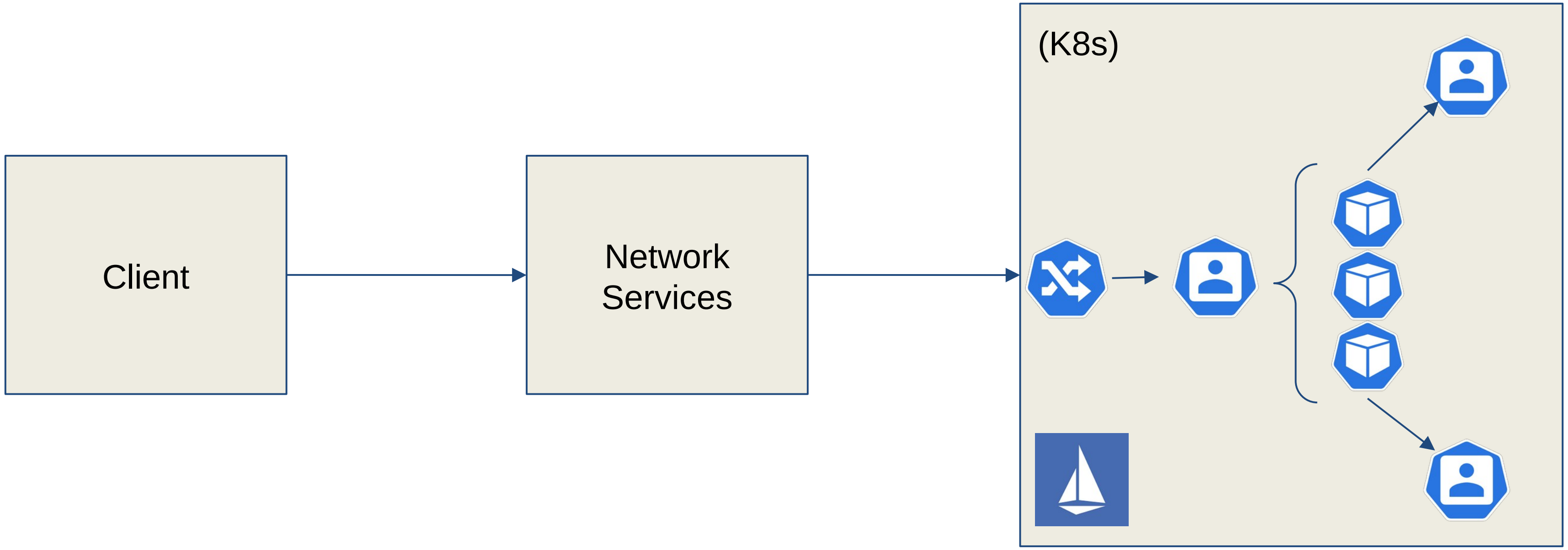
Where do we need to be secure?

Calling other Services



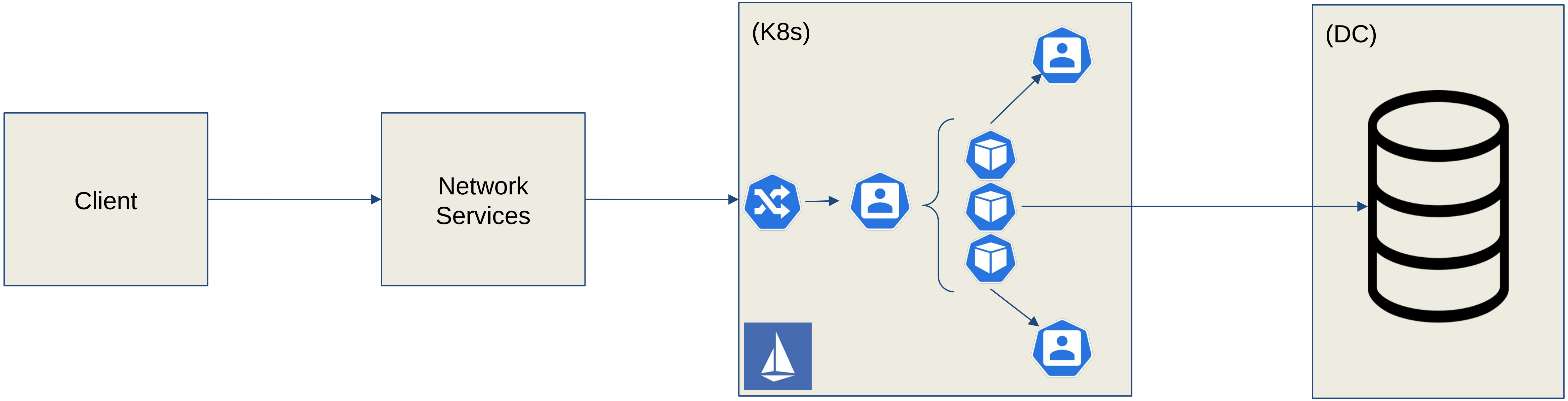
Where do we need to be secure?

Service Mesh

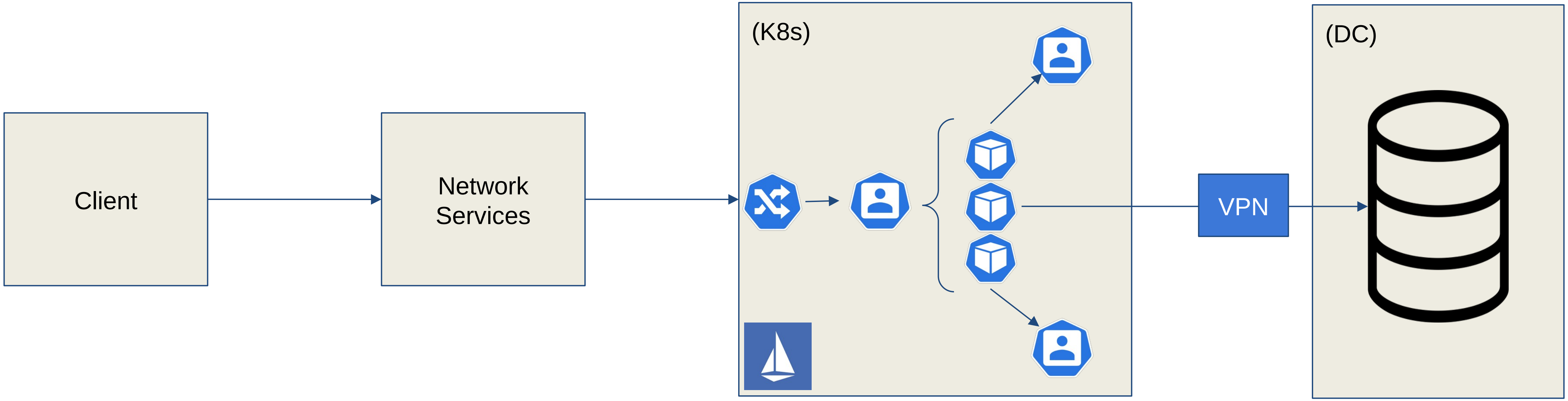


Where do we need to be secure?

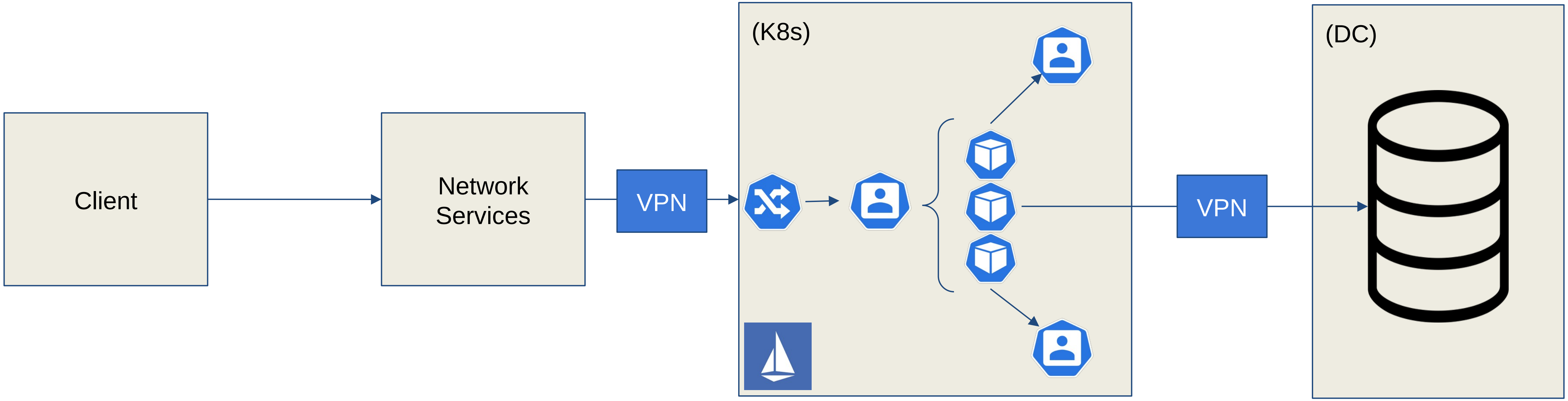
Hybrid Cloud / On-Prem



Where do we need to be secure?

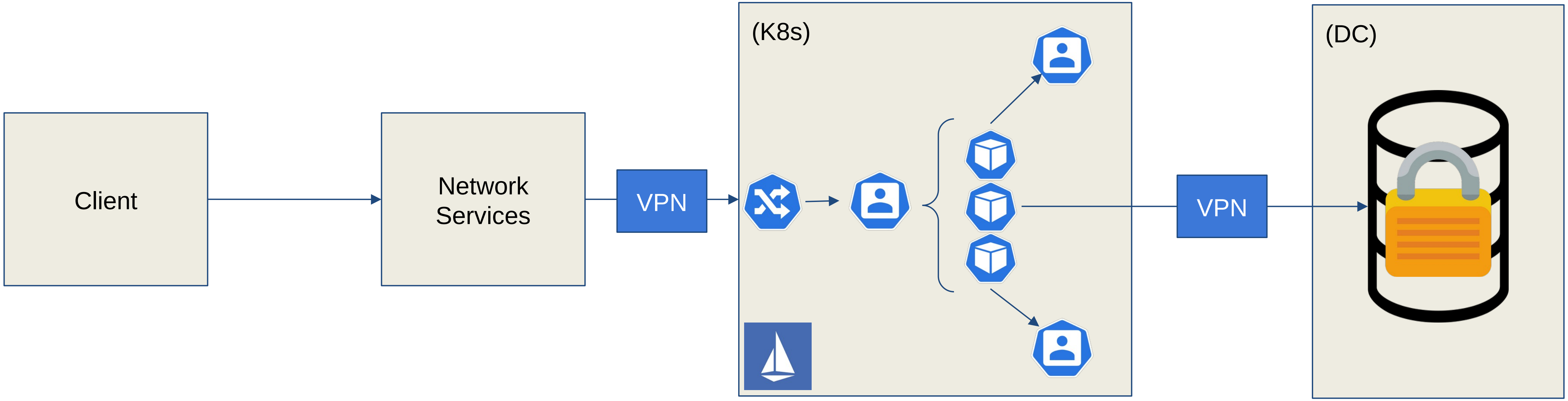


Where do we need to be secure?

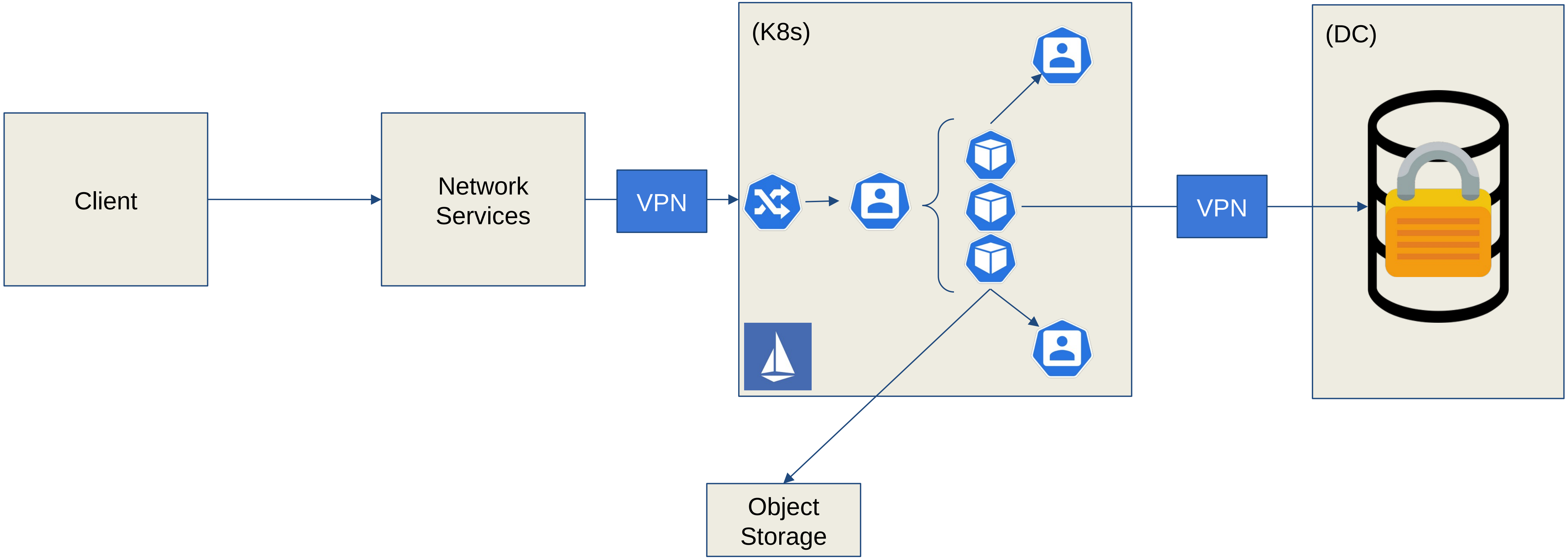


Where do we need to be secure?

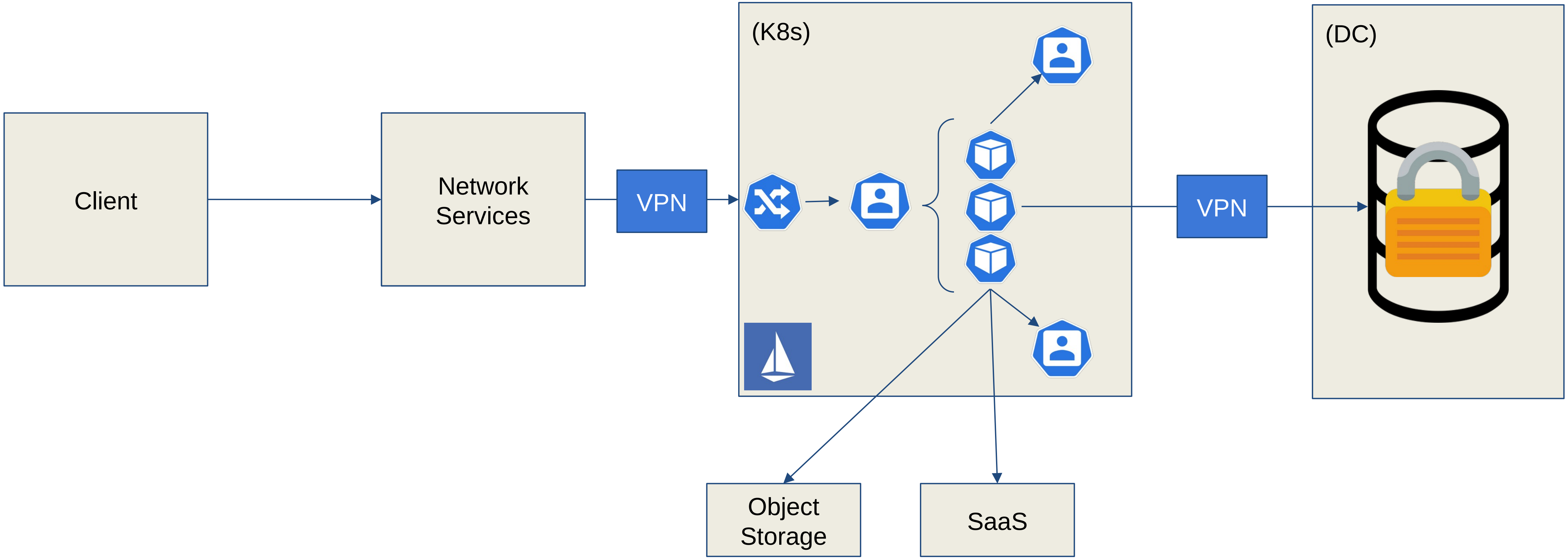
Secure in Motion and At Rest



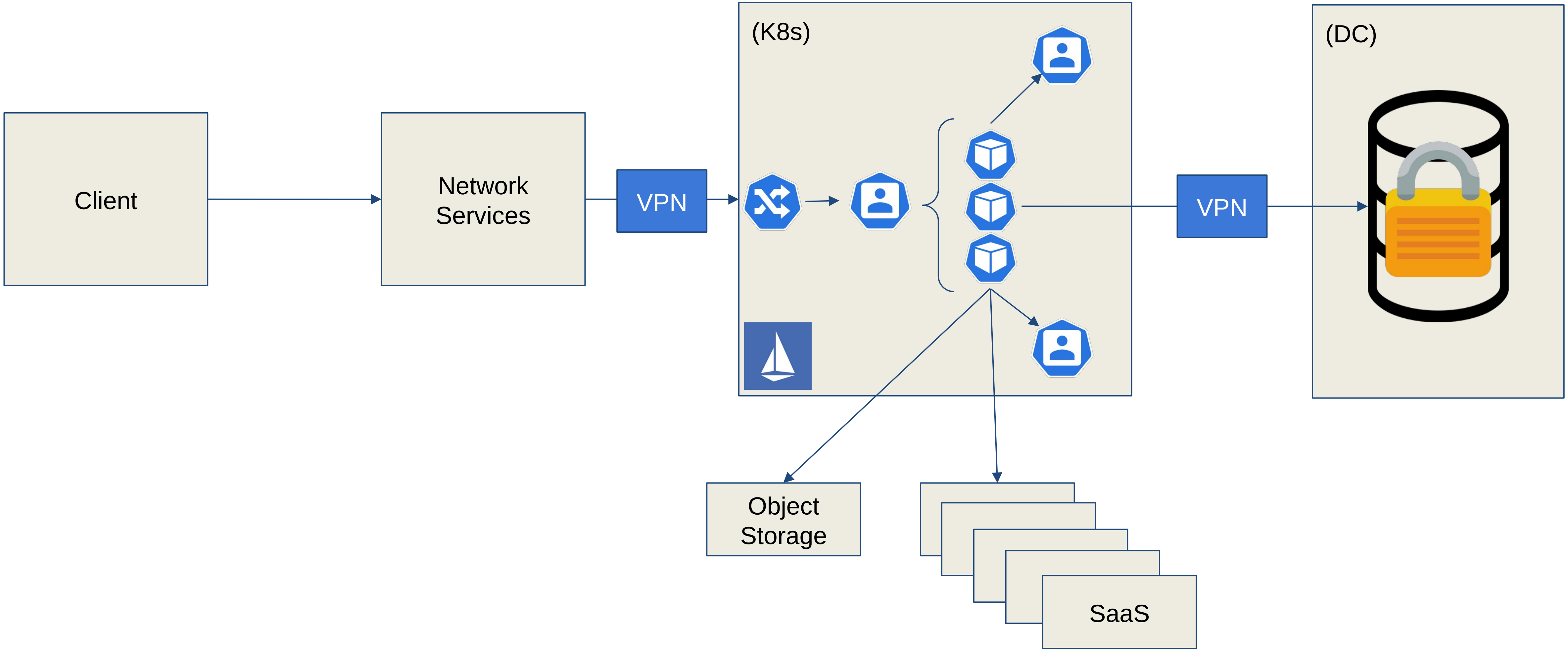
Where do we need to be secure?



Where do we need to be secure?

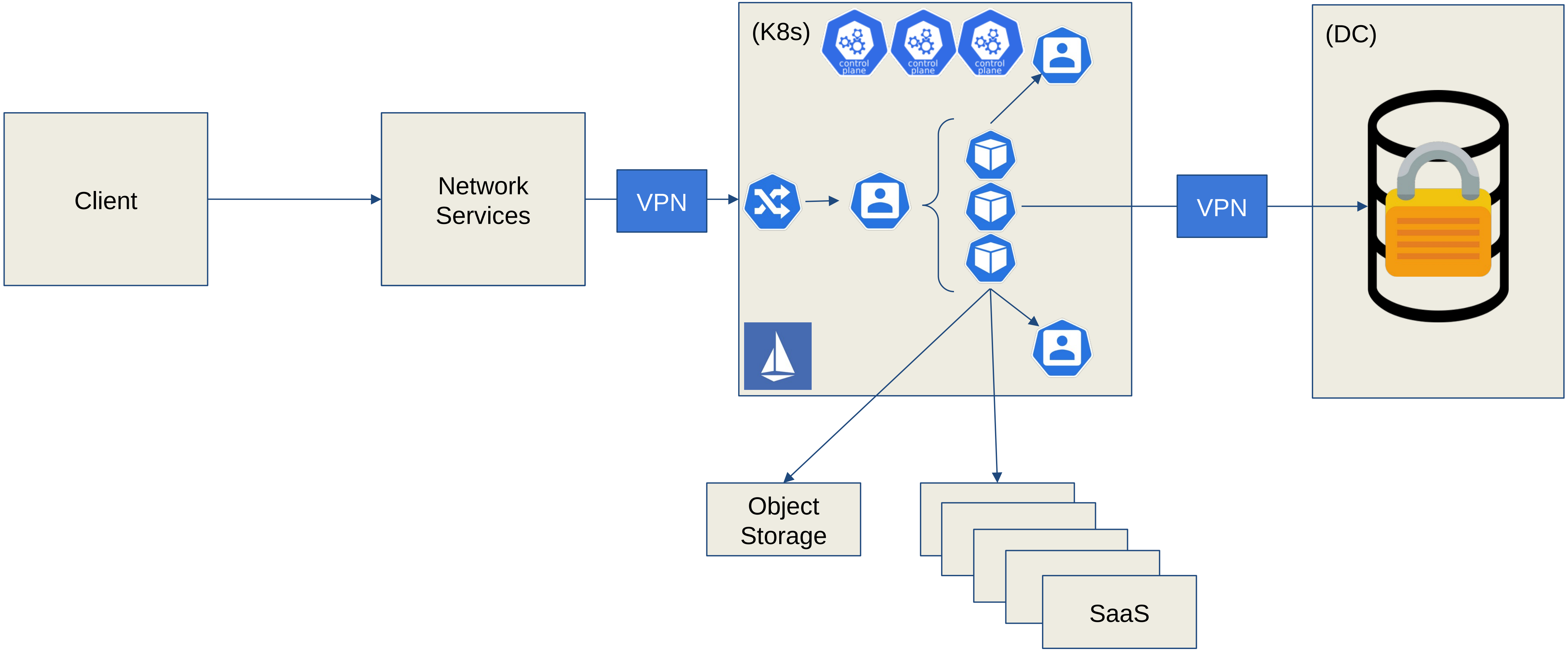


Where do we need to be secure?



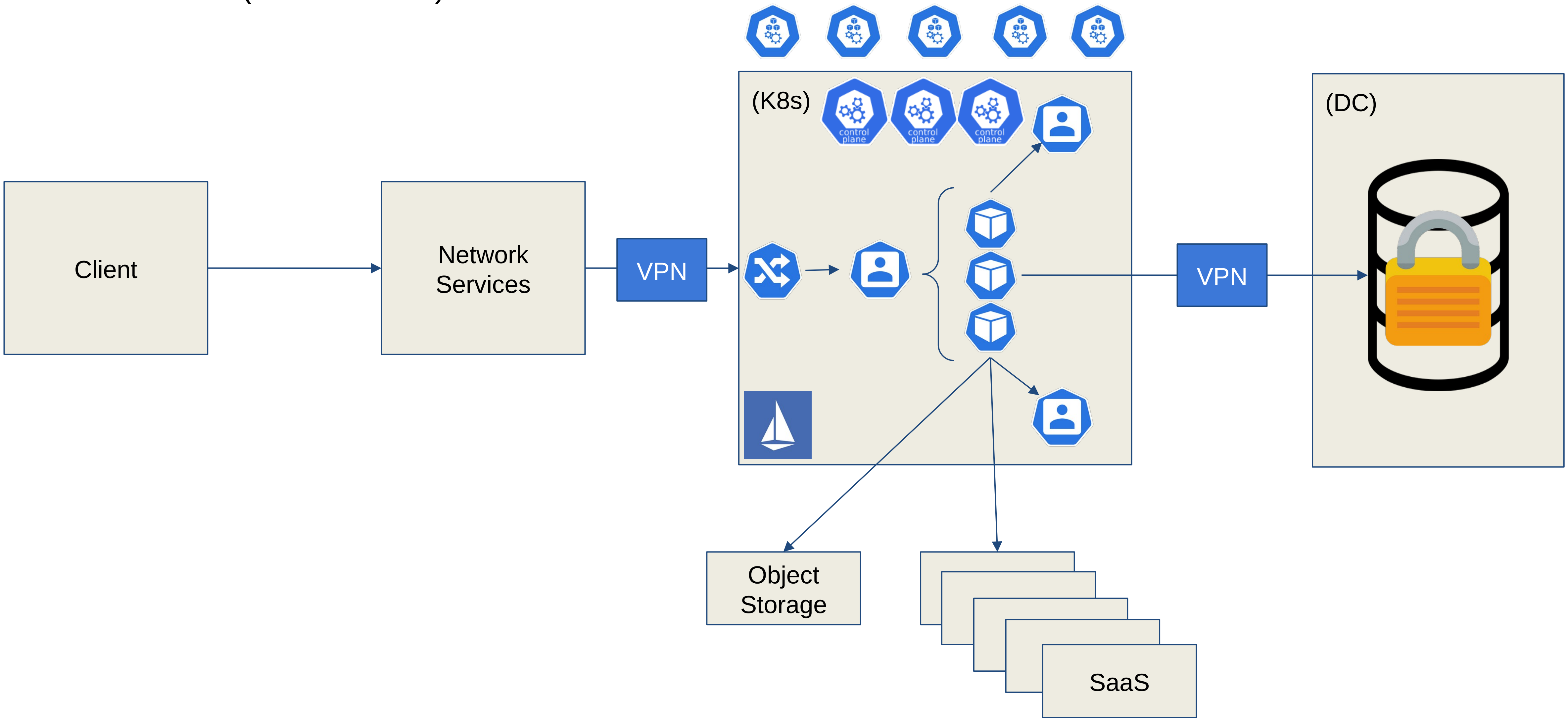
Where do we need to be secure?

Kubernetes Control Plane

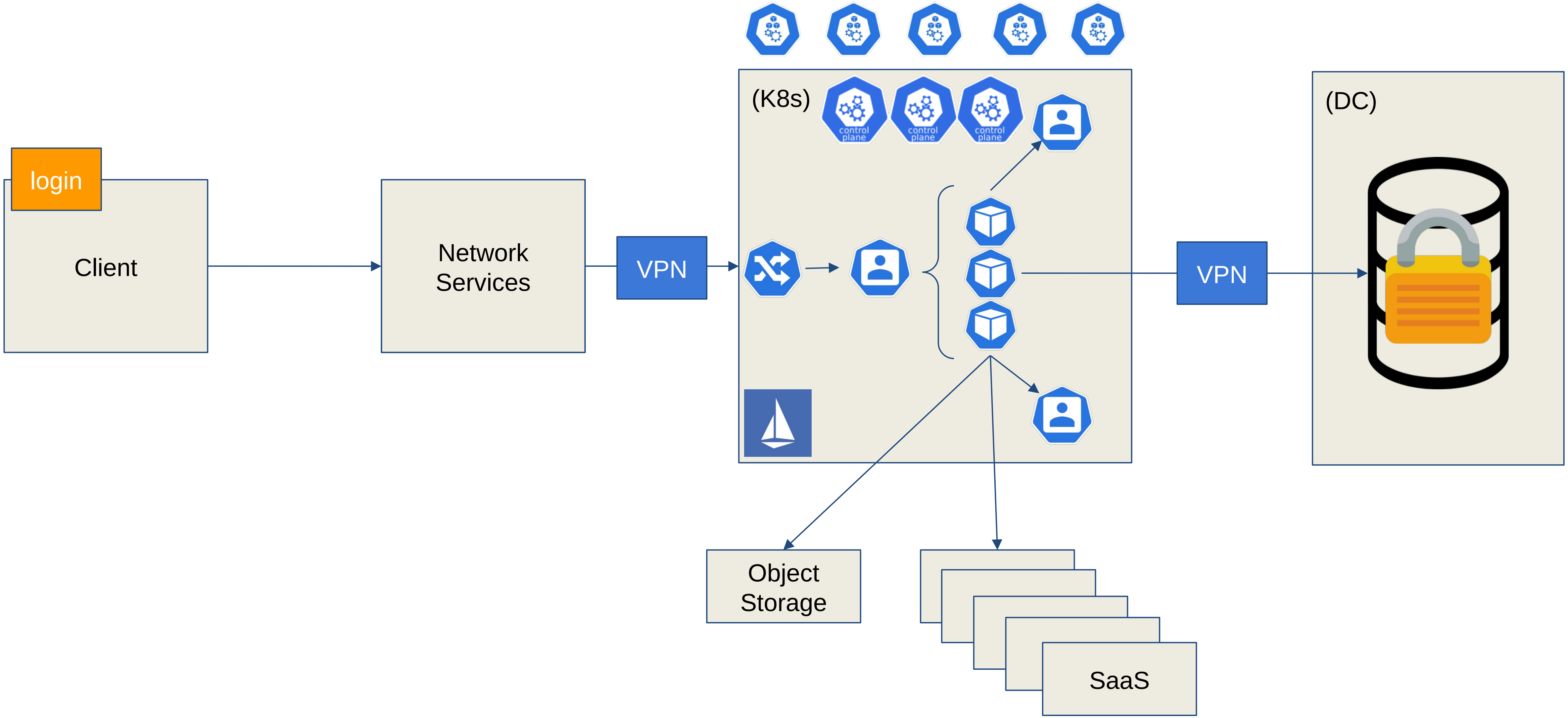


Where do we need to be secure?

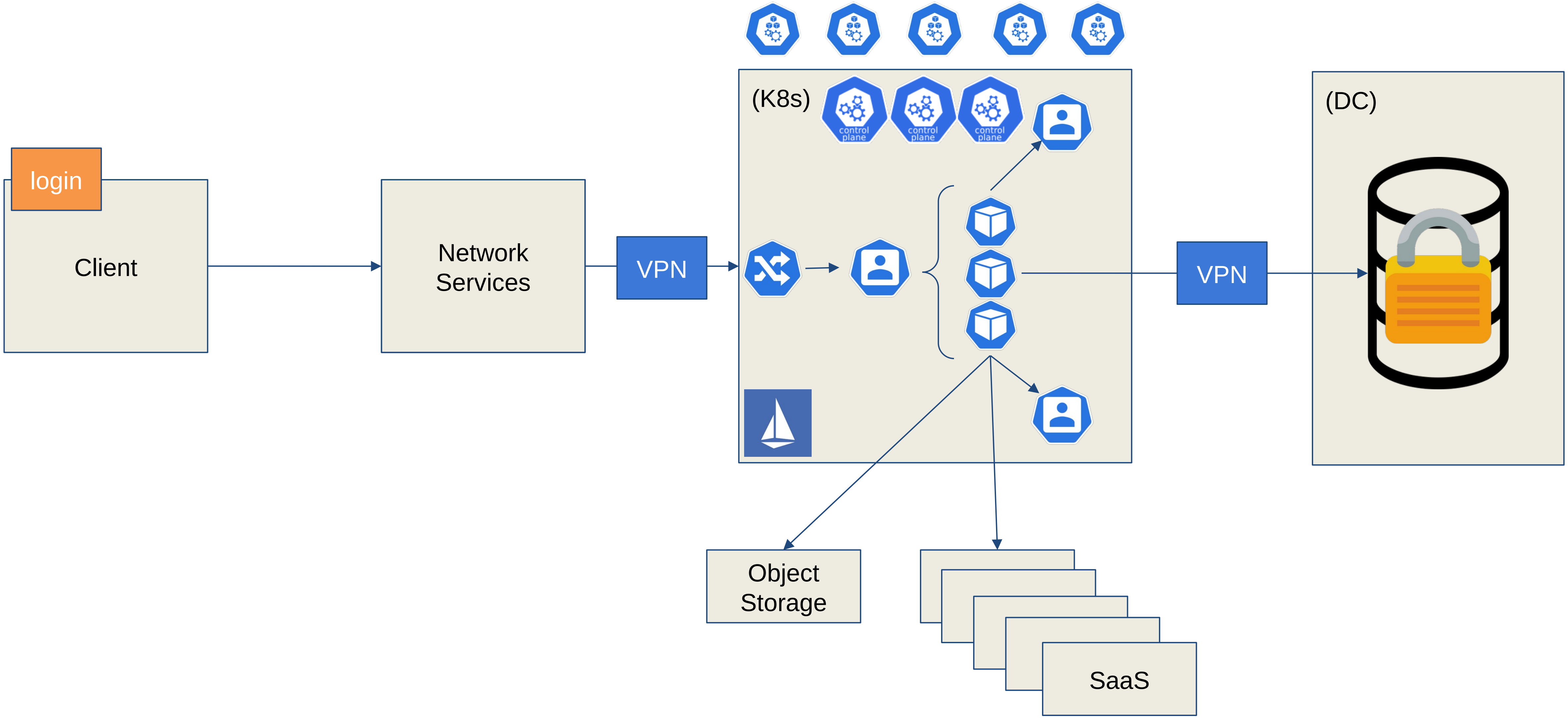
Kubernetes Nodes (and Volumes)



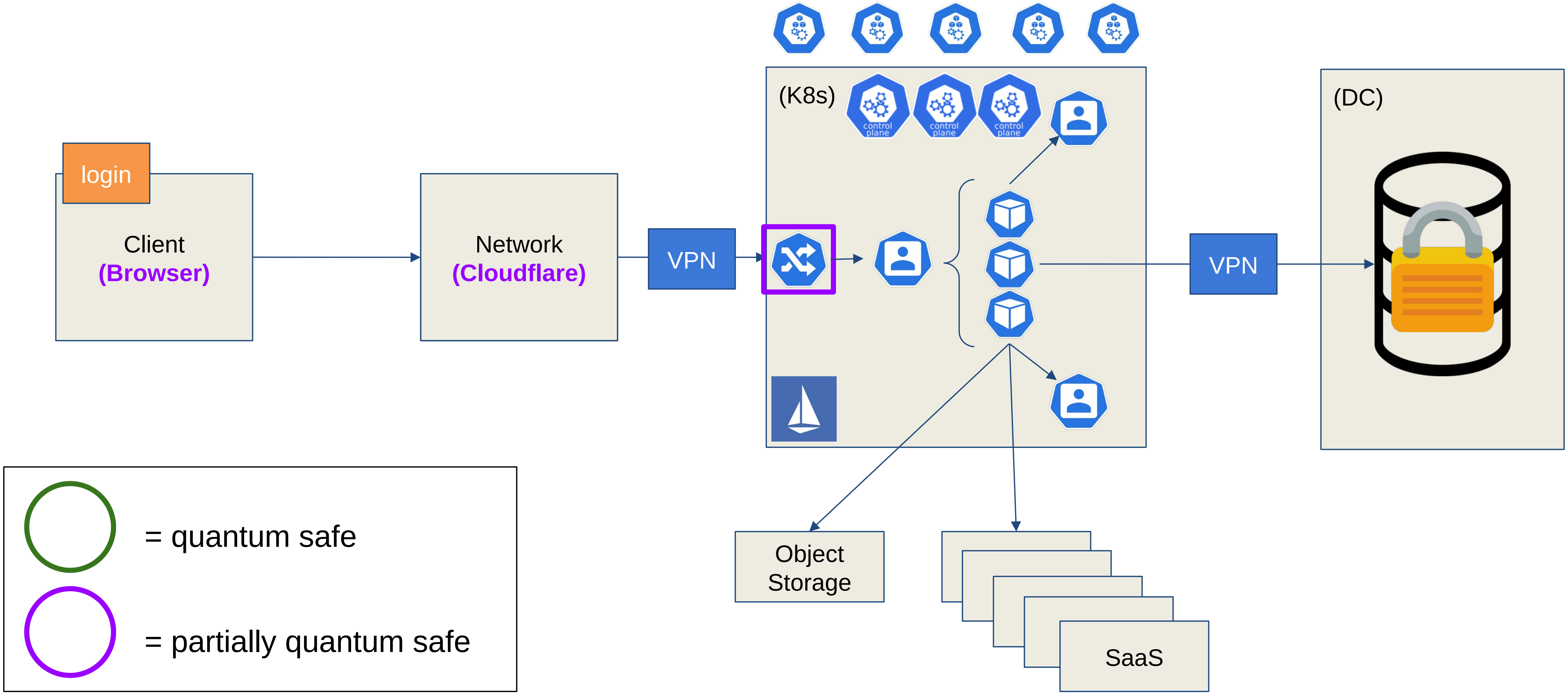
Where do we need to be secure?



Where ARE we secure?



Where ARE we secure?



First steps

- Cloudflare
  - https://blog.cloudflare.com/post-quantum-for-all/
- Chromium
  - https://blog.chromium.org/2023/08/protecting-chrome-traffic-with-hybrid.html
- Go
  - https://github.com/open-quantum-safe/liboqs-go
  - https://github.com/cloudflare/go
- OpenSSL
  - https://www.openssl.org/blog/blog/2023/10/26/ossl-32-beta/
- OpenSSH
  - https://github.com/open-quantum-safe/openssh
- OS
  - https://packages.fedoraproject.org/pkgs/liboqs/liboqs/
  - https://launchpad.net/ubuntu/+source/liboqs
  - https://tracker.debian.org/pkg/liboqs
  - https://lists.freebsd.org/pipermail/dev-commits-ports-main/2021-September/018107.html

What’s Next?

- Discover: inventory cryptographic assets
- Observe: stay informed of new standards and vulnerabilities
- Transform: swap existing crypto for quantum-safe crypto agility

Please scan the QR Code above
to leave feedback on this session

KubeCon | CloudNativeCon

North America 2023