

Understandable Microservices

...with  Jolie

Fabrizio Montesi <fmontesi@imada.sdu.dk>



DEPARTMENT OF MATHEMATICS
AND COMPUTER SCIENCE

Microservices

Autonomous services that run independently and communicate by message passing.

Microservices

Autonomous services that run independently and communicate by message passing.

- Microservices are components.
- They have independent lifecycles (start, stop).
- They are concurrent.

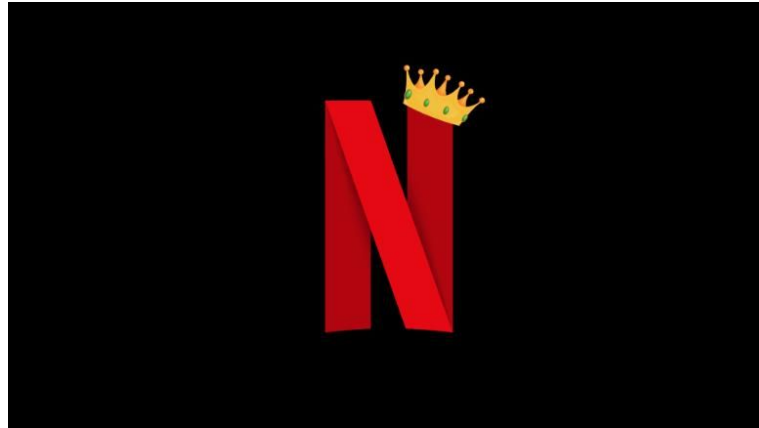
Why Microservices? (A few examples)

You can replicate them.



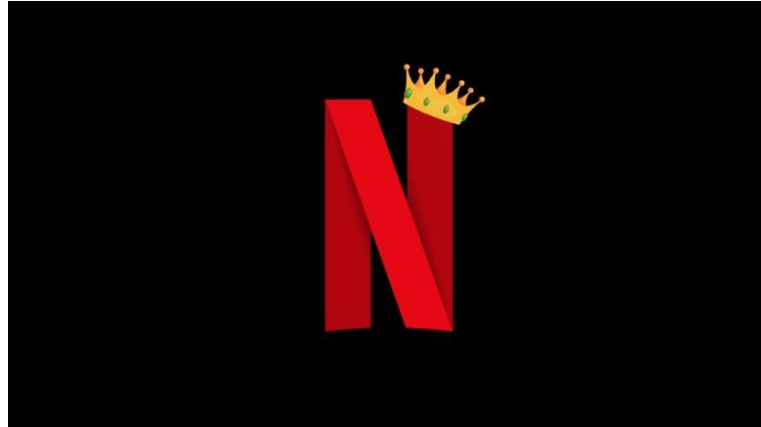
Why Microservices? (A few examples)

You can replicate them.



Why Microservices? (A few examples)

You can replicate them.
(If you program them right.)



Why Microservices? (A few examples)

Independent lifecycles make updates easier.



Why Microservices? (A few examples)

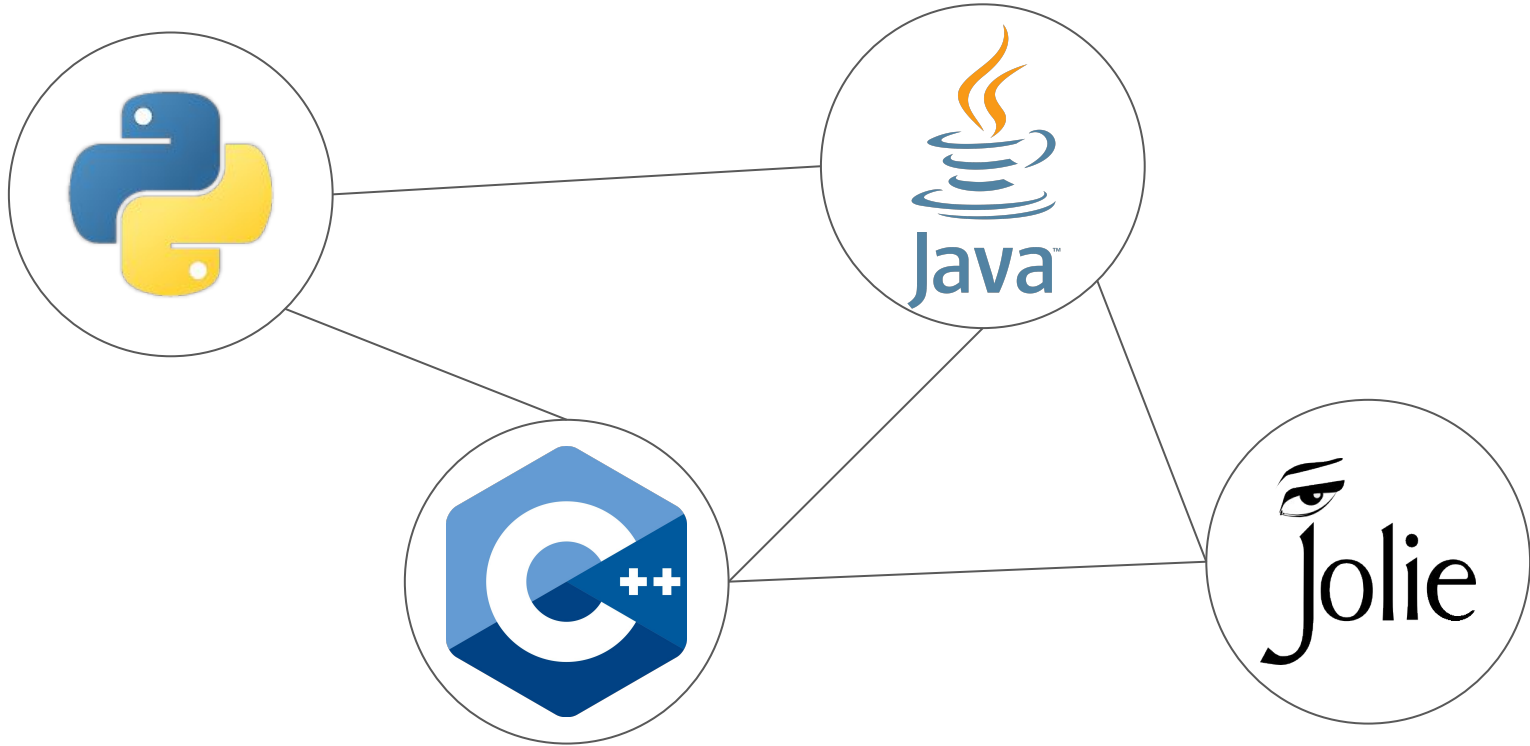
They are “technology-agnostic”.



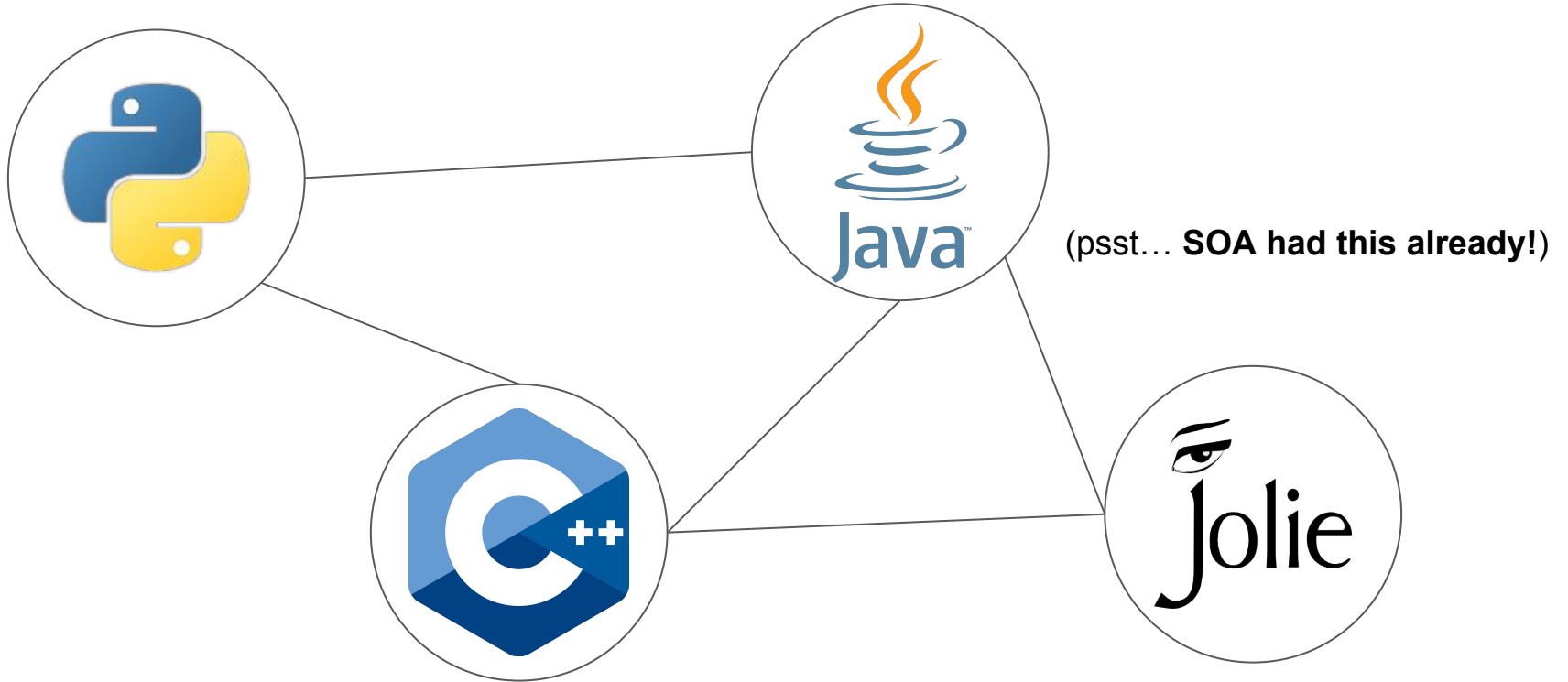
...



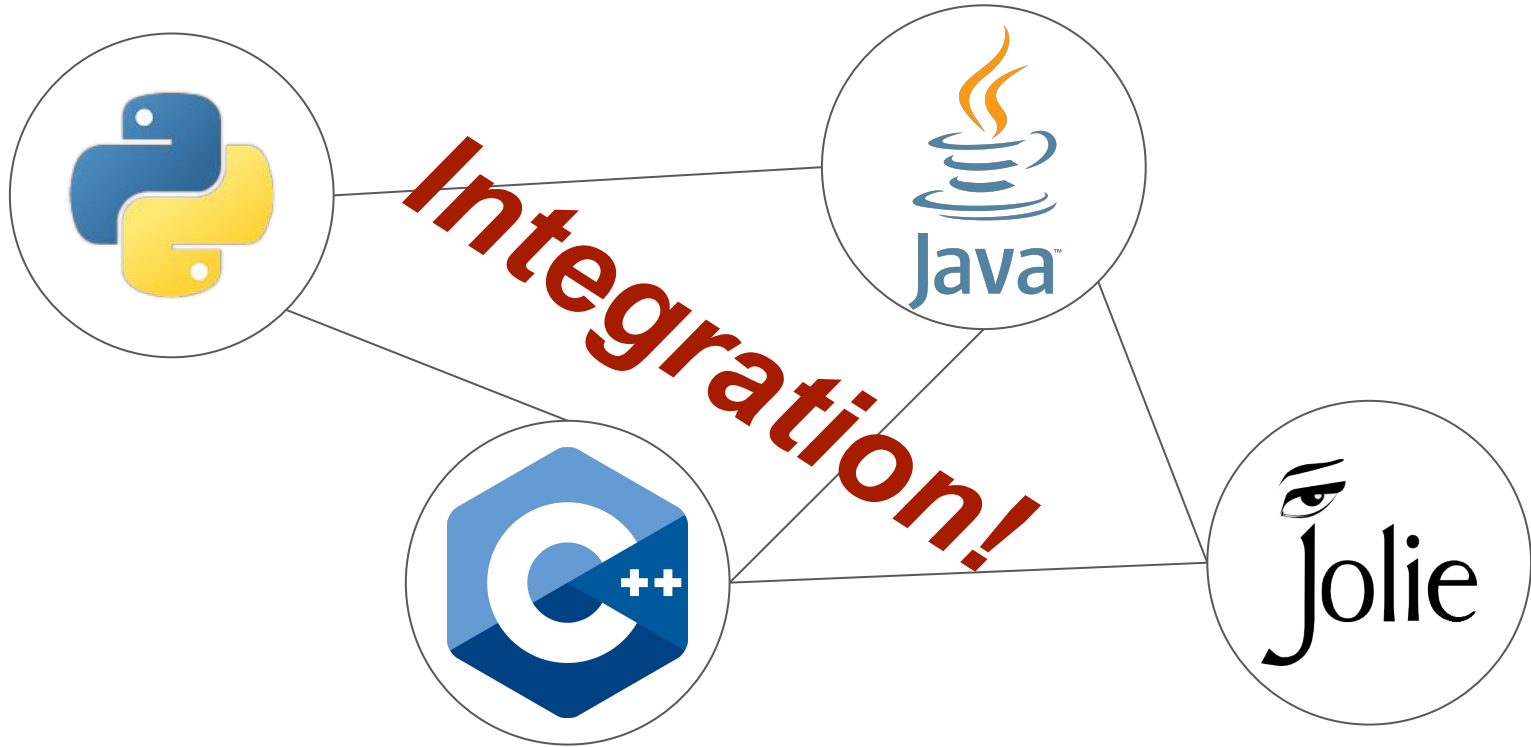
A Microservice system



A Microservice system

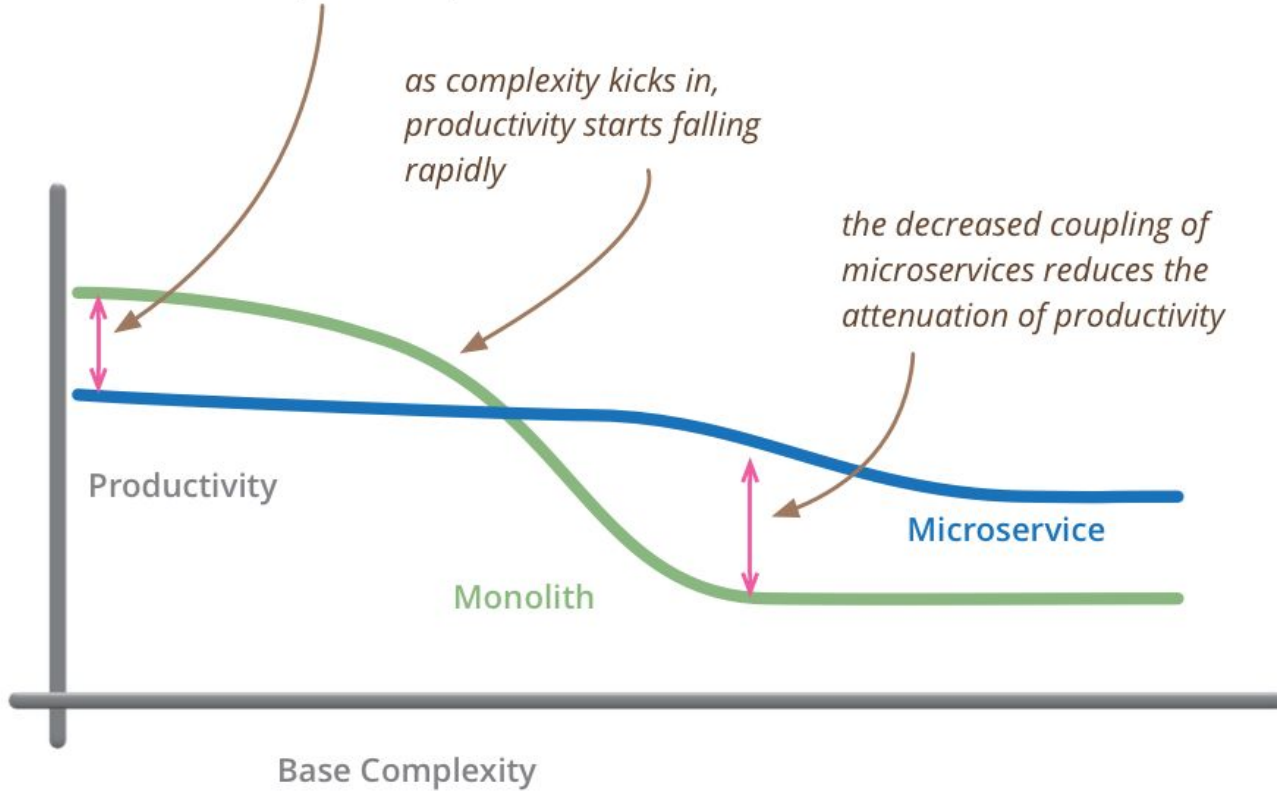


A Microservice system



[Fowler, 2015]

for less-complex systems, the extra baggage required to manage microservices reduces productivity



but remember the skill of the team will outweigh any monolith/microservice choice

The people: “Adopt Microservices if you...”

- “...need to scale.”
- “...have really complicated software.”

The people: “Adopt Microservices if you...”

- “...need to scale.”
- “...have really complicated software.”

Conclusion:

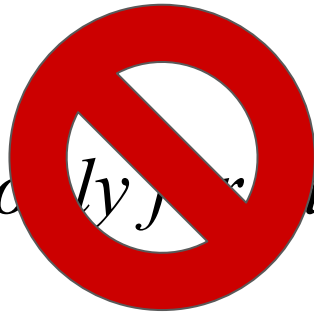
“Use Microservices only for big businesses/stuff.”

The people: “Adopt Microservices if you...”

- “...need to scale.”
- “...have really complicated software.”

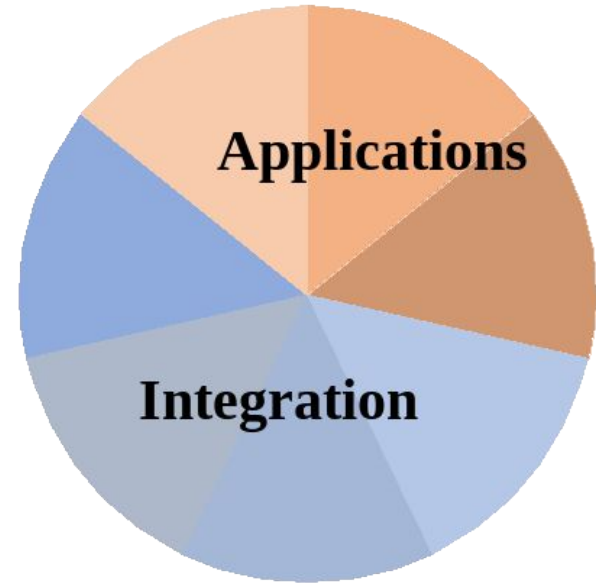
Conclusion:

“Use Microservices only for big business stuff.”



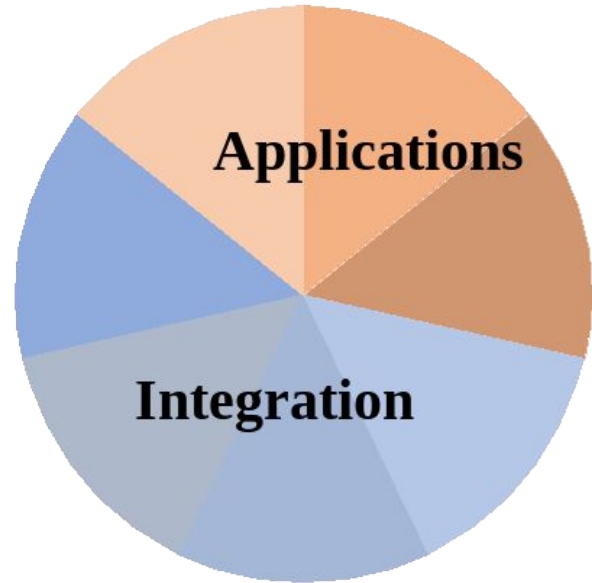
Integration costs more than applications

- Approximately $\frac{1}{3}$ more
[Forrester Research]
[Gartner]



Integration costs more than applications

- Approximately $\frac{1}{3}$ more
[Forrester Research]
[Gartner]
- Complexity is widespread.



Sundhedsplatform for Region Sjælland

- ~150 million EUR invested in 2016, then re-estimated ~375 million EUR.

**Ekstra
Bladet**

35 overlæger advarer: IT-system giver patientkaos

- Den største udfordring er de tekniske problemer, dvs. integration af Det Fælles Medicinkort, hvor man ordinerer medicin for patienterne, og systemet, hvor man bestiller blodprøver. Og så er der nogle udfordringer omkring at ordinere kemoterapi for

Jolie: a microservice-oriented language

Jolie: a microservice-oriented language

Why a language?


Jolie: a microservice-oriented language

Why a language?


Languages influence how we think.

Jolie: a microservice-oriented language

Jolie: a microservice-oriented language

- Nice logo:  Jolie

Jolie: a microservice-oriented language

- Nice logo: The logo for Jolie features a stylized eye above the word "Jolie" in a serif font.

- Innovative:



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE



INRIA

Centre de recherche BOPHIA ANTIPOLIS - MÉDITERRANÉE



European Institute of
Innovation & Technology

italianaSoftware 

SDU 
UNIVERSITY OF
SOUTHERN DENMARK



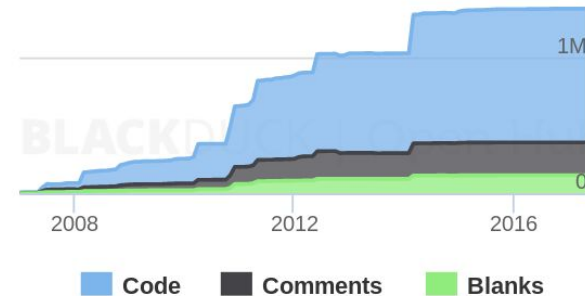
Jolie: a microservice-oriented language

- Nice logo: 


- Innovative:



- Open source: <https://jolie-lang.org>



Jolie: a microservice-oriented language

- Nice logo: 

- Innovative:



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE



INRIA

SERIE DE RECHERCHE SOPHIA ANTIPOLIS - MÉDERRANÉE



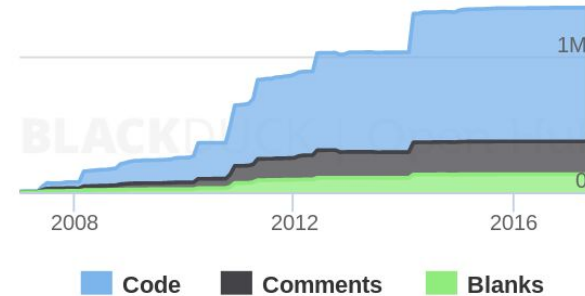
European Institute of
Innovation & Technology

italianaSoftware

SDU
UNIVERSITY OF
SOUTHERN DENMARK



- Open source: <https://jolie-lang.org>



Jolie: a microservice-oriented language

- The weapon of choice of:
 - integration ninjas;
 - wise software designers.



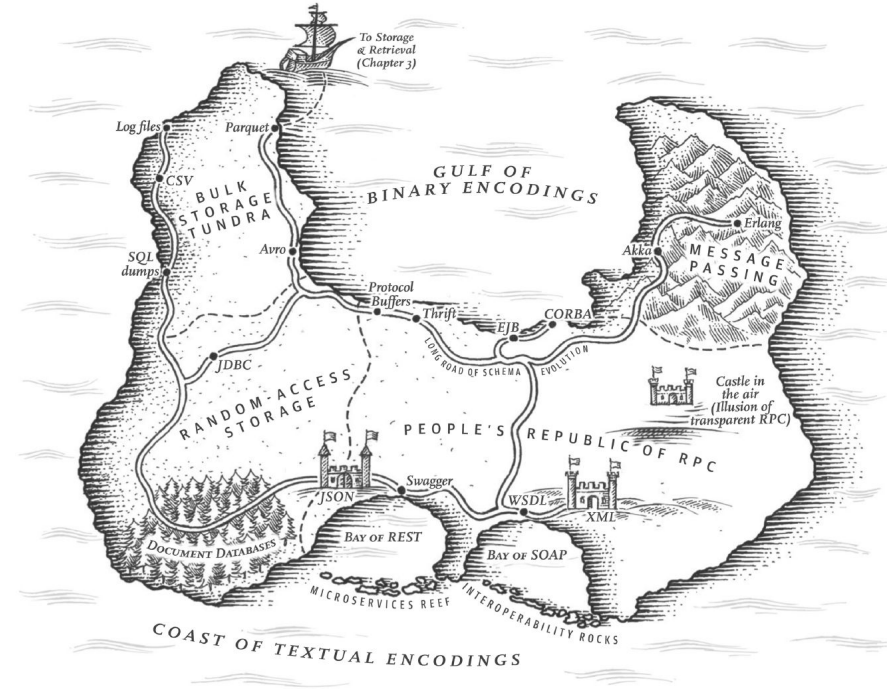
Jolie: a microservice-oriented language

- The weapon of choice of:
 - integration ninjas;
 - wise software designers;
 - unwise software designers!

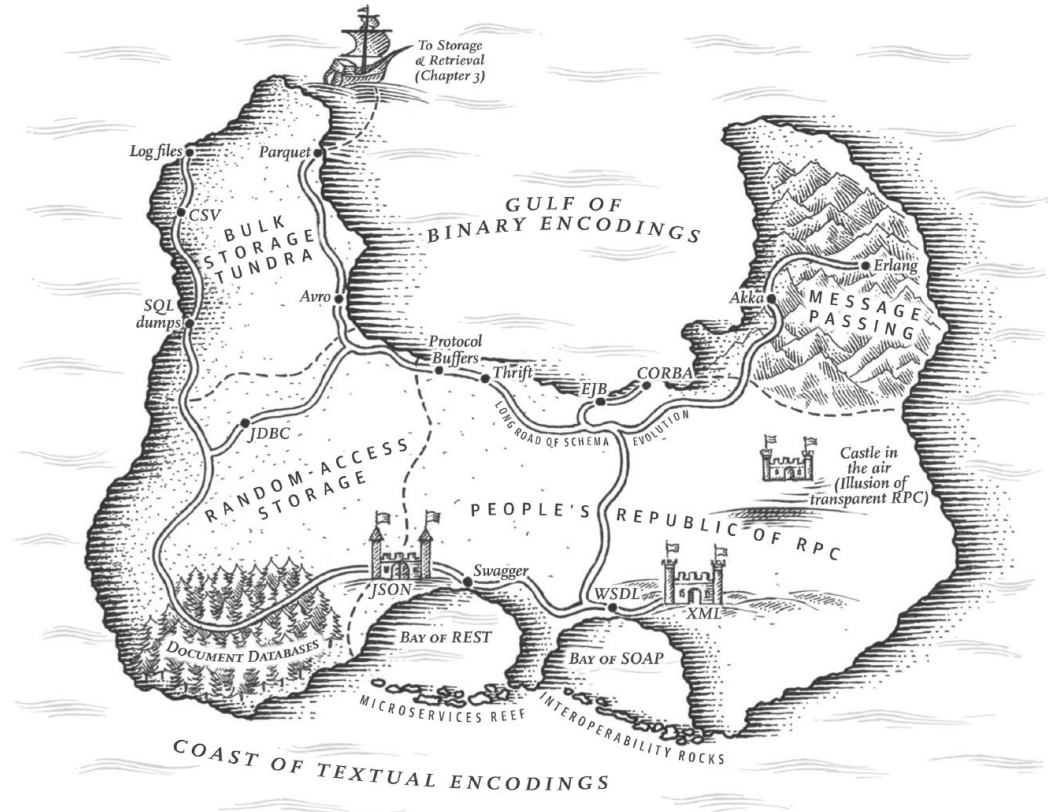


Checklist for composing components (services)

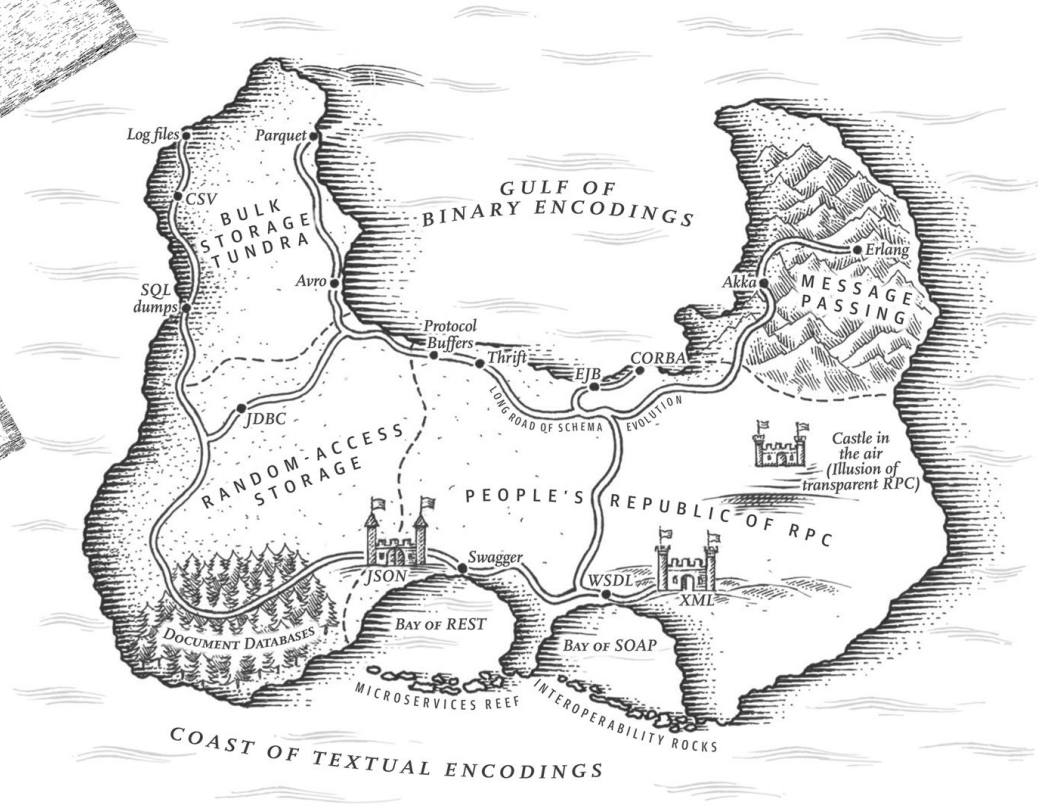
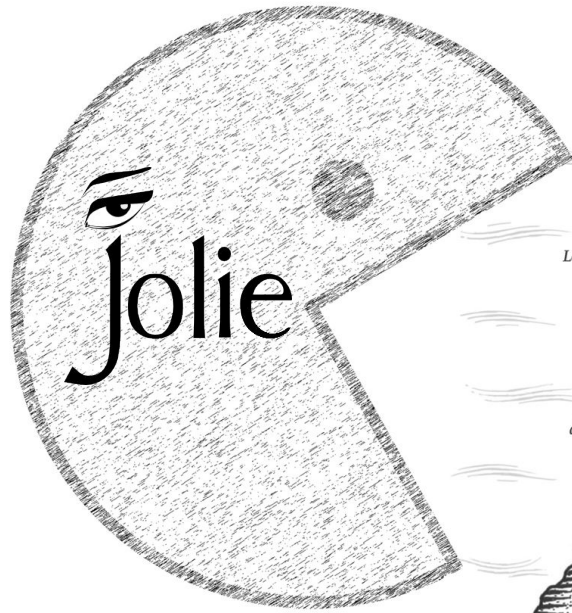
- Communications.
- Data formats.
- APIs.
- Behavioural integration.
- Access points.



(Credits: Kleppman, 2017)



(Credits: Kleppman, 2017)



(Credits: Giallorenzo, 2018)

Hello, world!

Hello, world!

```
include "console.iol"
```

```
main
```

```
{
```

```
    println@Console( "Hello, world!" )()
```

```
}
```

Include from standard library

```
include "console.iol"
```

```
main
```

```
{  
}  
}
```

```
println@Console( "Hello, world!" )()
```

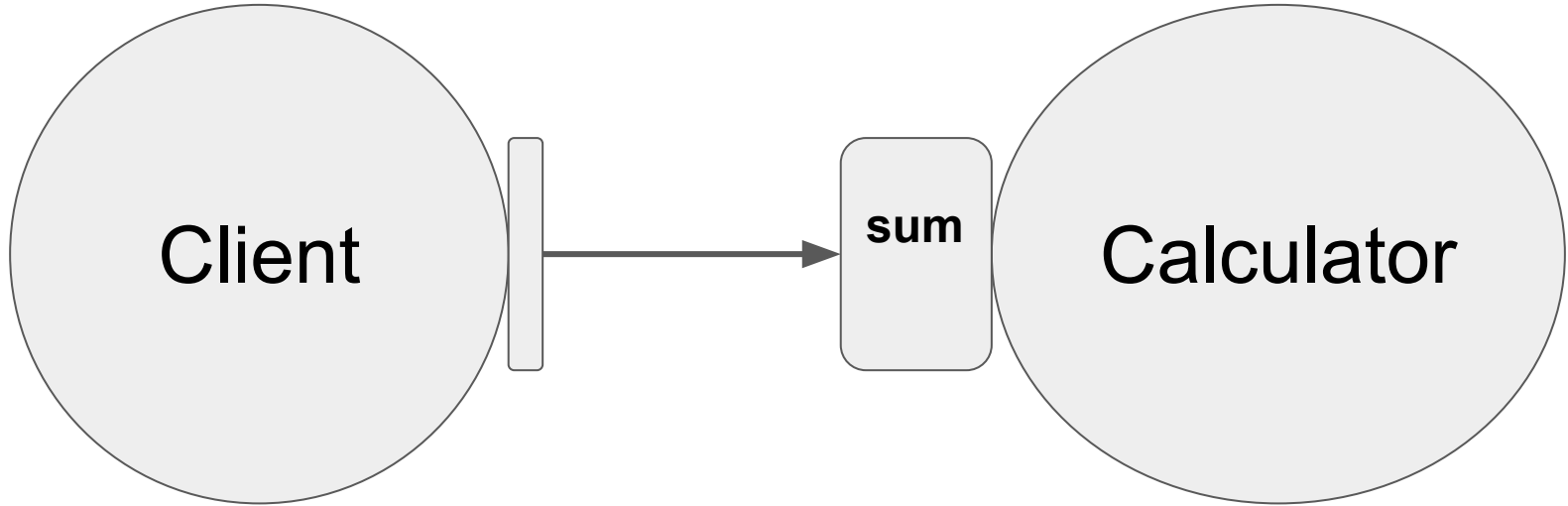
Program entry point

Operation

The service I want to invoke

All components are services

A calculator service



APIs

- APIs are defined as **interfaces**.
 - Example: “The *Calculator* interface consists of an operation *sum*, which receives two integers and replies with an integer”.
- **Interfaces** are exposed on **ports**.
 - Example: “The *Calculator* interface is available at localhost:8000 using the binary protocol SODEP”.

Protocols and Data Formats

- You can combine different protocols and data formats.

TCP/IP sockets

Unix sockets

Bluetooth

...

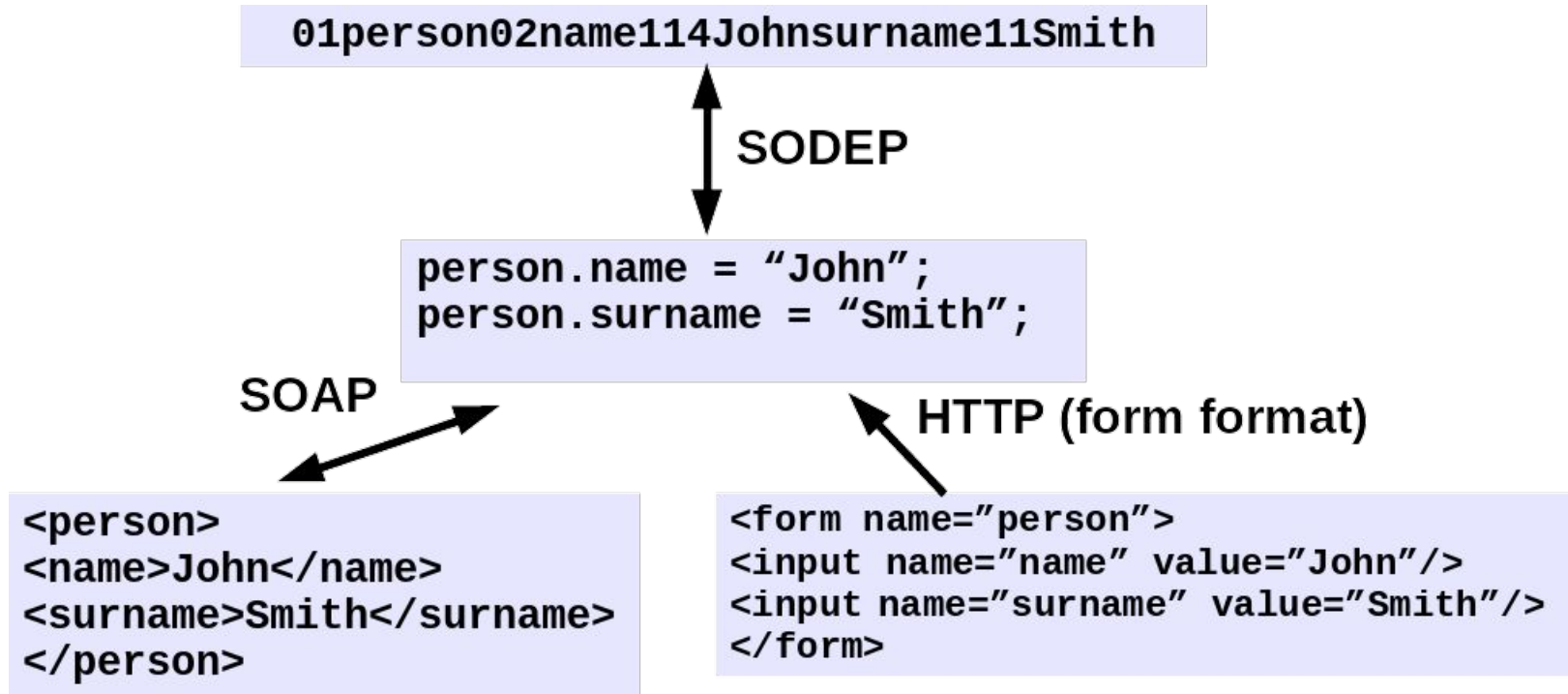
SODEP

SOAP

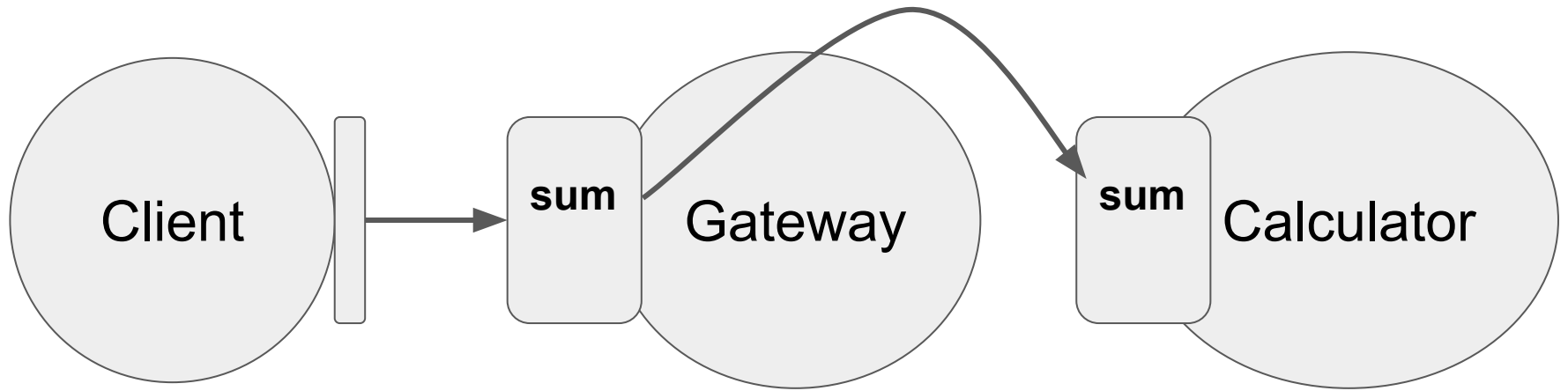
HTTP

...

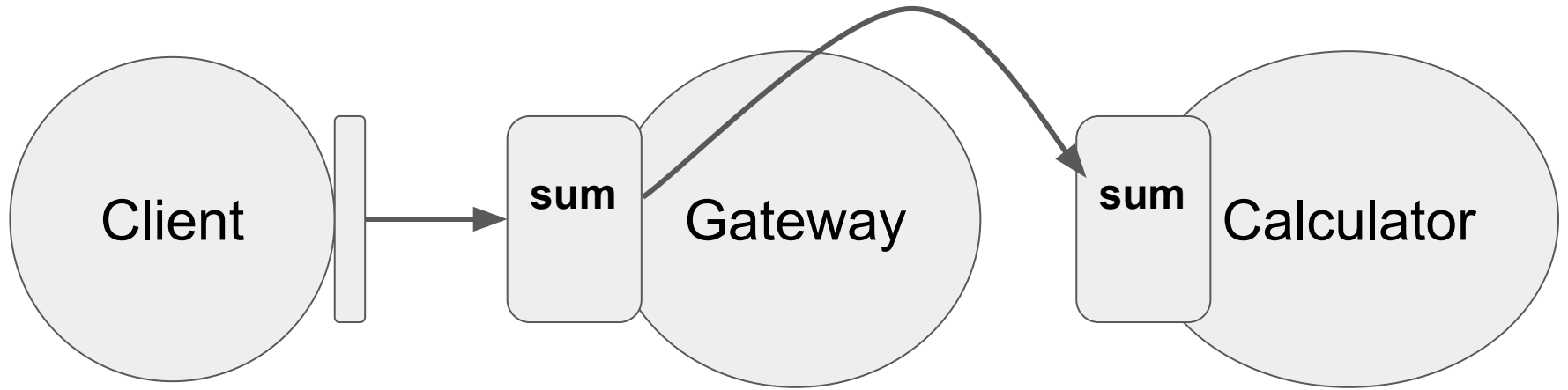
Automatic data conversion



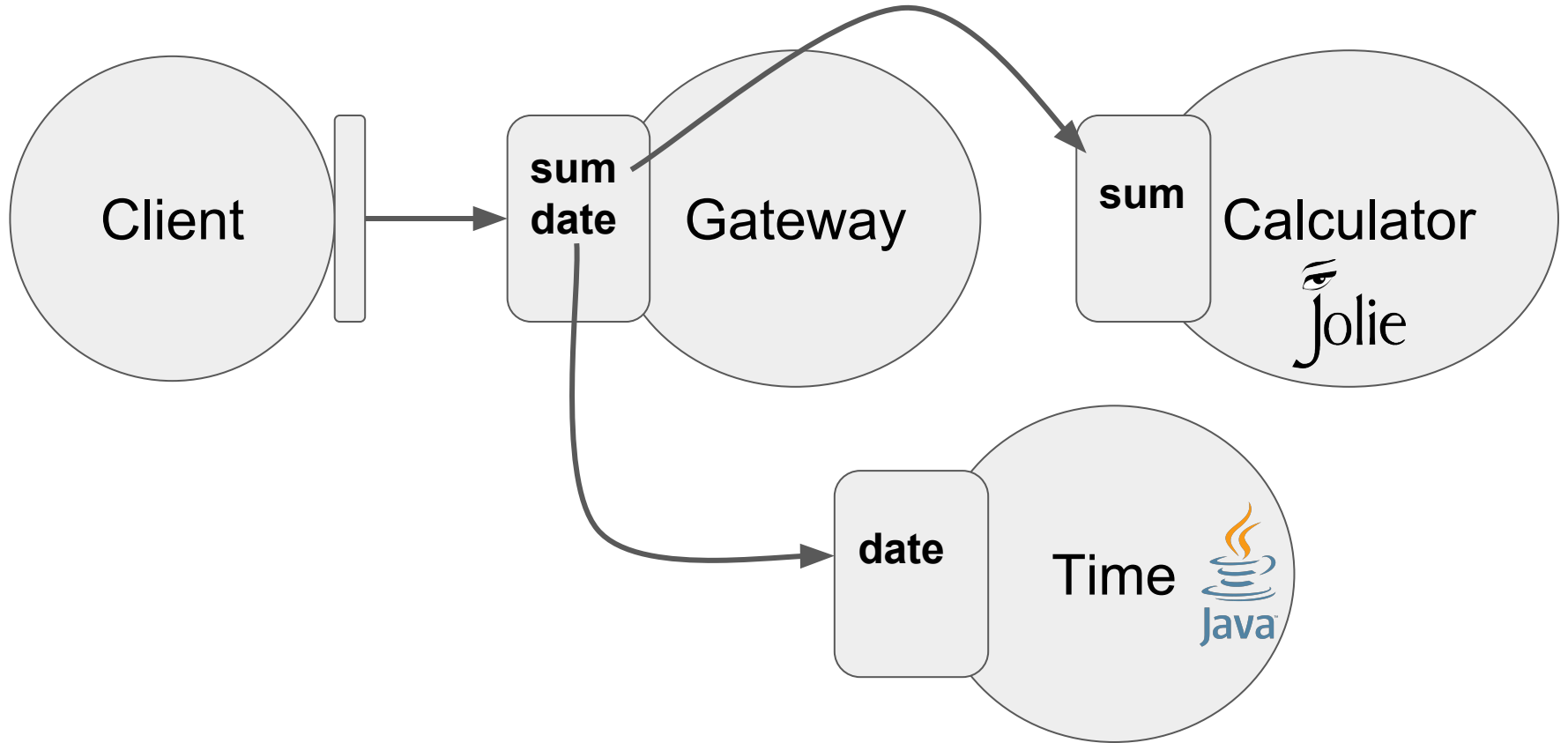
An API Gateway



An API Gateway (Embedding)



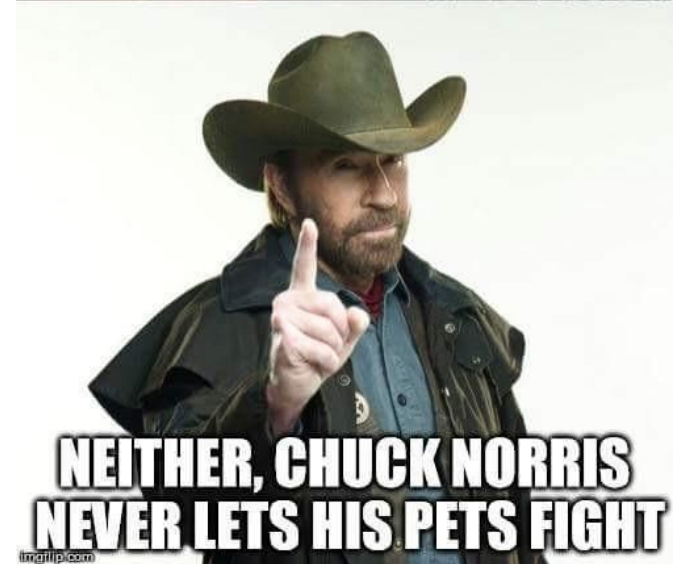
An API Gateway



A sophisticated example

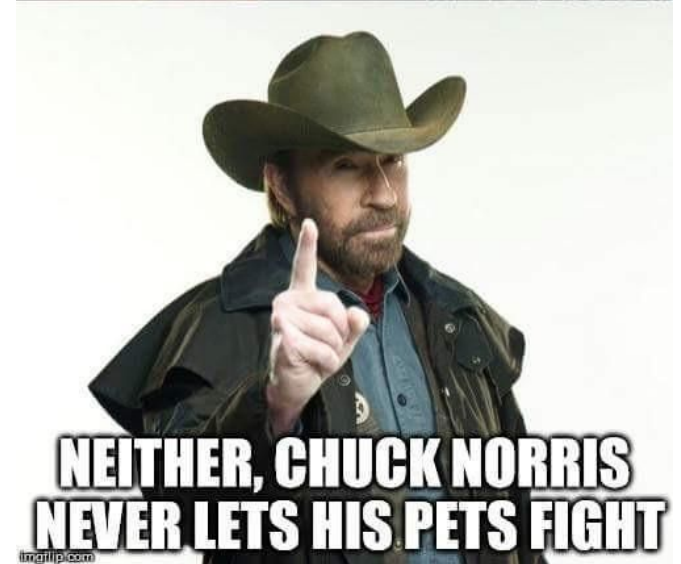
A sophisticated example

- Business case: a platform for sharing Chuck Norris jokes.



A sophisticated example

- Business case: a platform for sharing Chuck Norris jokes.
- Services available:



A sophisticated example

- Business case: a platform for sharing Chuck Norris jokes.
- Services available:
 - jokes from chucknorris.io;



A sophisticated example

- Business case: a platform for sharing Chuck Norris jokes.
- Services available:
 - jokes from chucknorris.io;
 - [telegra.ph](https://t.me/telegra.ph) for sharing.



Try the demo online

- <https://fmontesi.github.io/2018/08/16/jo.html>
- The source:
<https://github.com/fmontesi/jo-demo-chuck>
- On docker:
 - `docker pull fmontesi/jo-demo-chuck`
 - `docker run -it --rm -p 8080:8080
fmontesi/jo-demo-chuck`

Other useful native features

- Service discovery.
- Sessions.
- Distributed Fault and Compensation Handling.

Some successful application areas

- Digitalisation.
- Integrated business processes.
- E-Commerce.
- Internet of Things.

The future

- Automatic replication.
- Automatic stateful -> stateless transformation.
- Support for more protocols and communication technologies.

Get in touch

- www.jolie-lang.org
- famontesi@gmail.com
- Skype: fmontesi
- Commercial support: www.italianasoftware.com