



Asynchronous in
microservices



Who we are?

Dawid Wiśniewski



Requirements

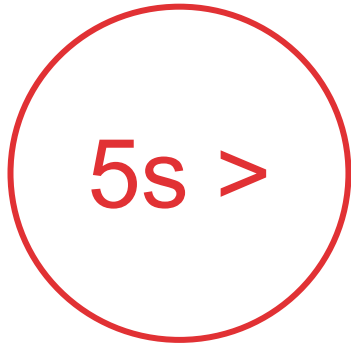


Reliability

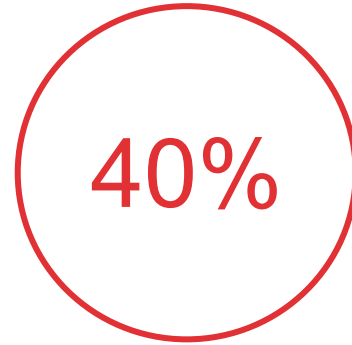


Performance

Research

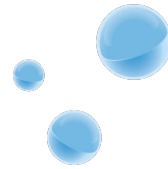


Delays



Conversion

Solutions

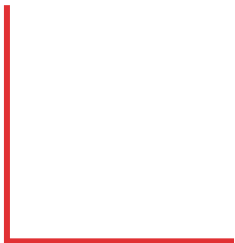


VARNISH
CACHE

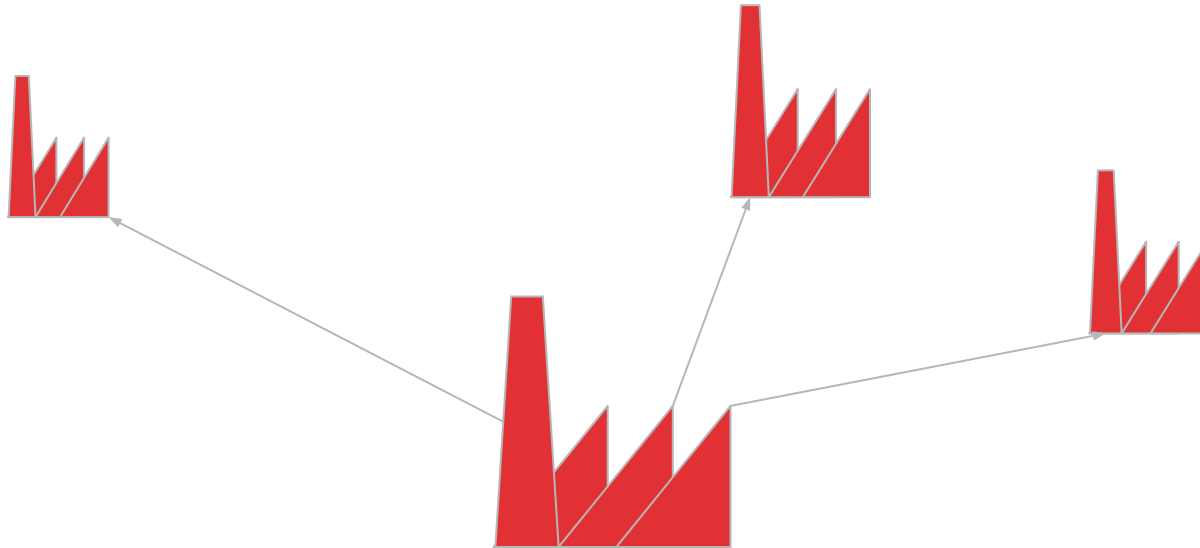


Load balancing

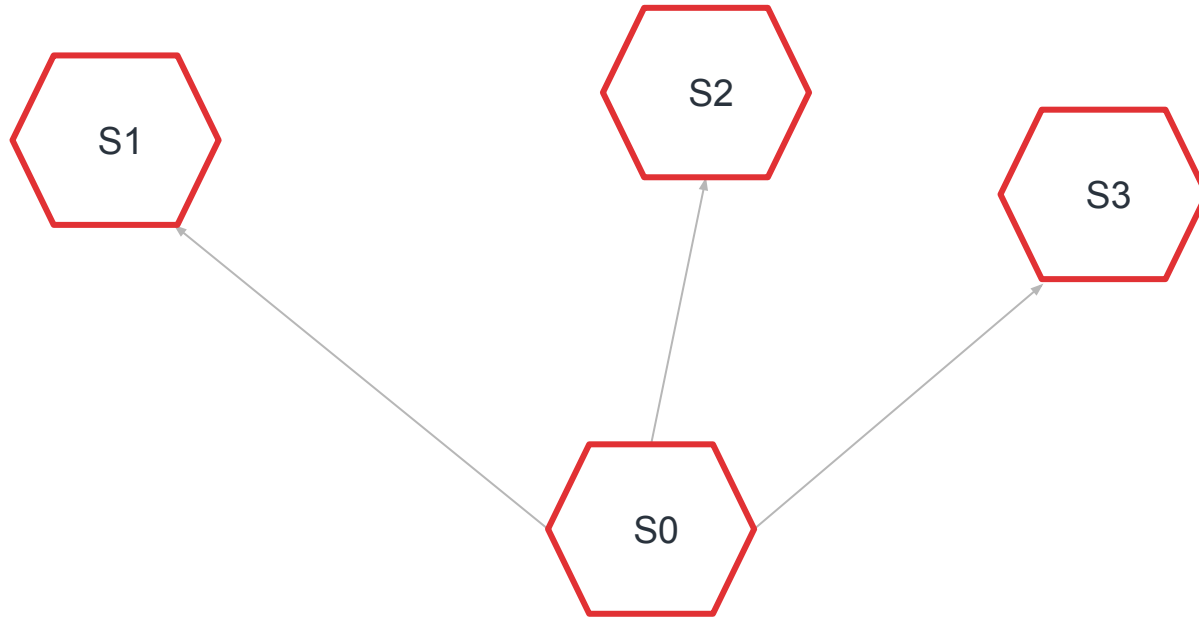
Microservices



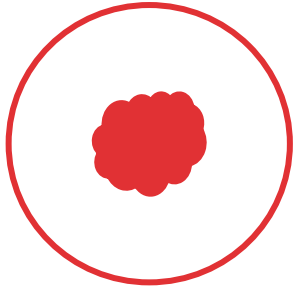
Example 1



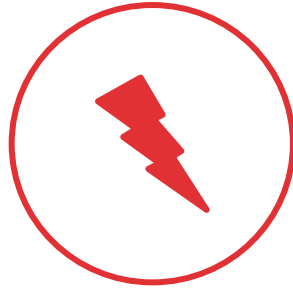
Example 2



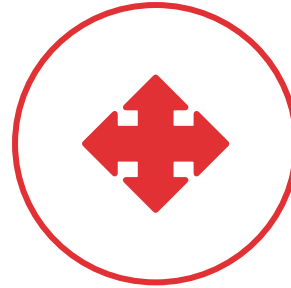
Benefits



Heterogeneity



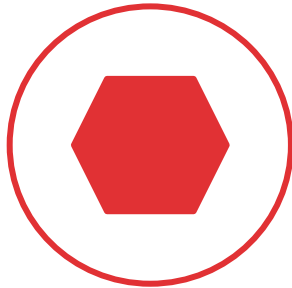
Fault tolerance



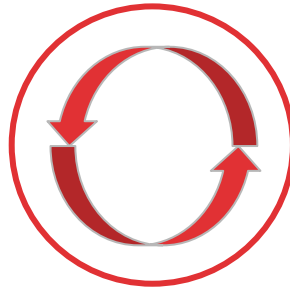
Scalability



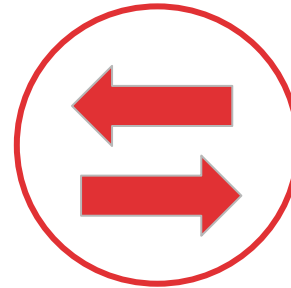
Ease of
implementation



Organization of teams



Interoperability

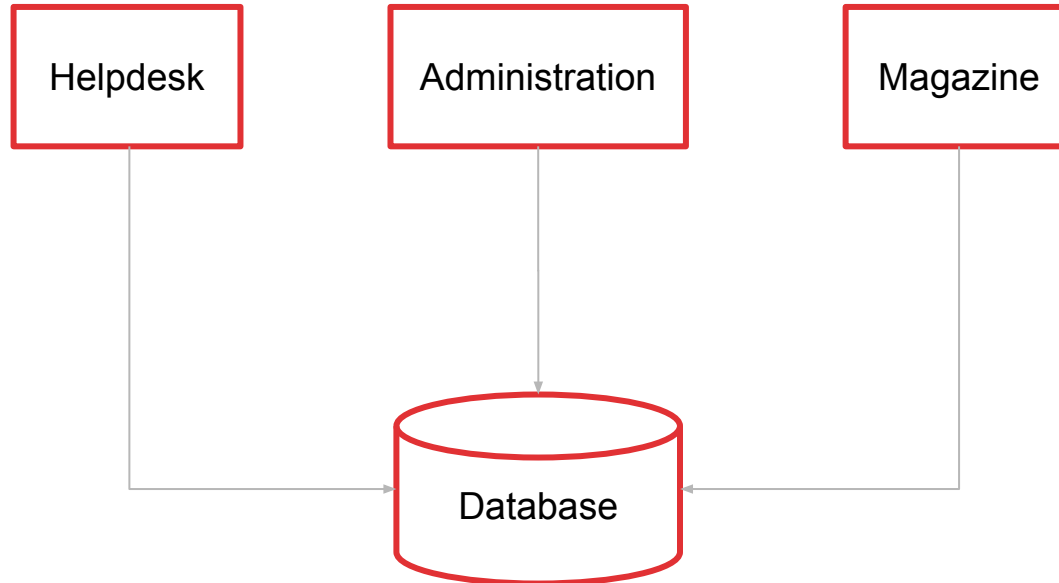


Exchangeability

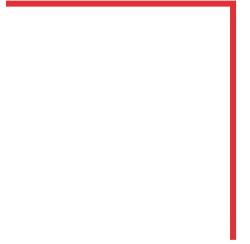
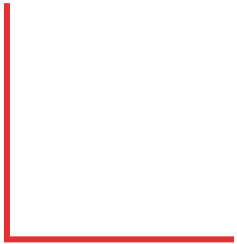
Challenges

- Integration
- Communication
- Prevent duplicates
- Versioning
- Monitoring

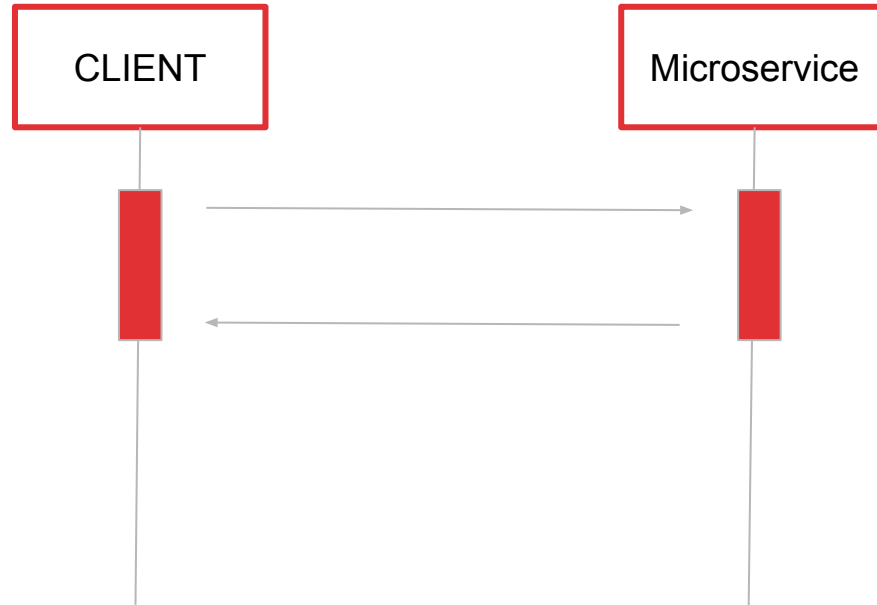
Integration by database



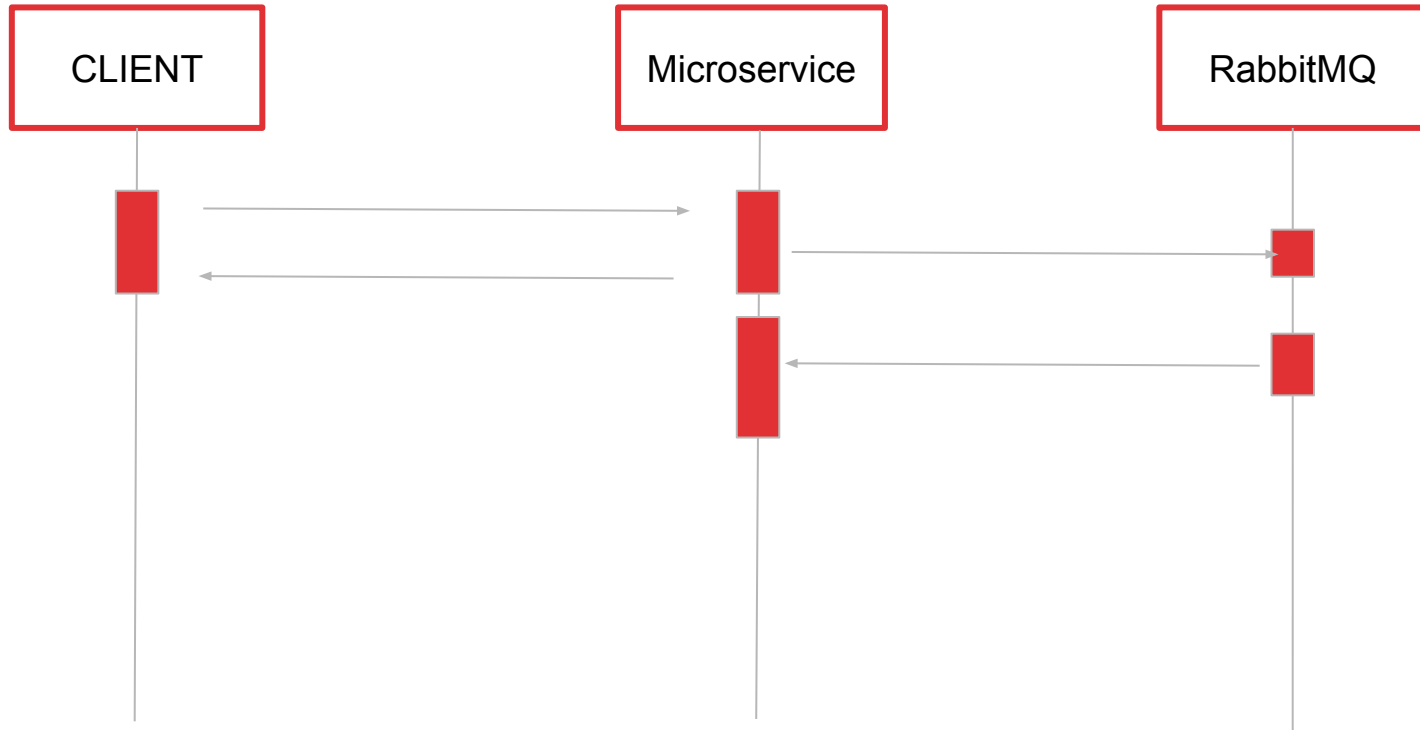
HTTP API



SYNCHRONOUS



ASYNCHRONOUS



JSON request

```
{  
  'request-id': `12345`  
  'email' : test@dialcom24.pl,  
  'name': 'Jan',  
  'surname': 'Kowalski',  
  'callbackUrl': 'https://example.pl/callbackUrl'  
}
```

Request

```
public function execute(Request $request)
{
    $response = new DispatchTaskResponse();

    try {
        $this->validator->validate($request);
        $task = new SimpleTask(Queues::ECHO_TEST, $request->toArray());
        $command = $this->commandFactory->createByData($task->getRequestArrayData());
        $this->commandRepository->save($command);
        $task->setTicketId($command->id);
        $response->setTicketId($command->id);
        $this->messageBroker->dispatch($task);

    } catch (Exception $ex) {
        $response->setErrorMessage($ex->getMessage());
    }

    $this->presenter->present($response);
}
```


Consumer

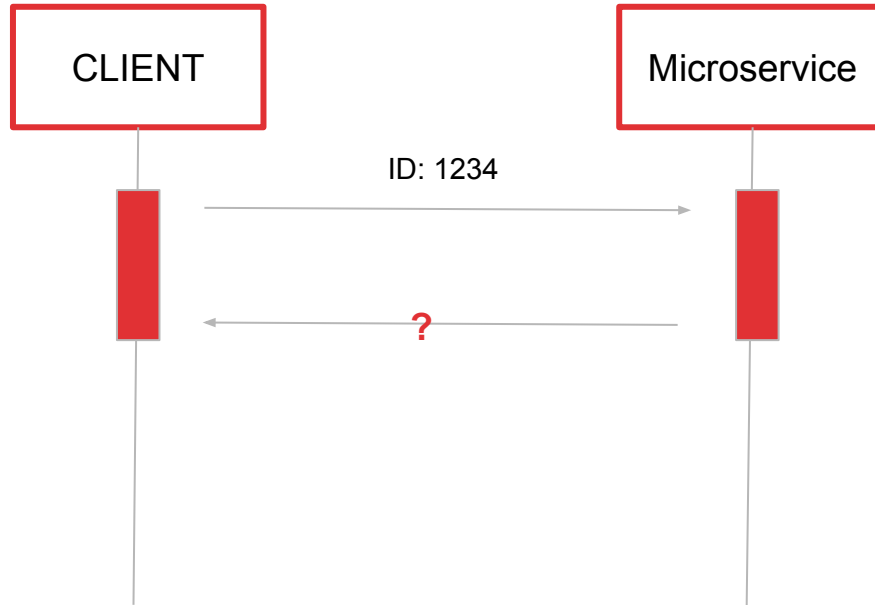
```
function execute(Message $message)
{
    try {
        $command = $this->findCommand($message);
        $this->handleUseCase($message, $command);
        $confirmed = $this->sendResponseToCallbackUrl($command);
        $this->saveCommand($command);
    } catch (\Exception $exception) {
        $confirmed = false;
        $this->logger->error('general exception occurred', [
            $message->getTicketId(),
            $message->getData(),
            __METHOD__,
            $exception->getMessage()
        ]);
    }

    return $confirmed;
}
```

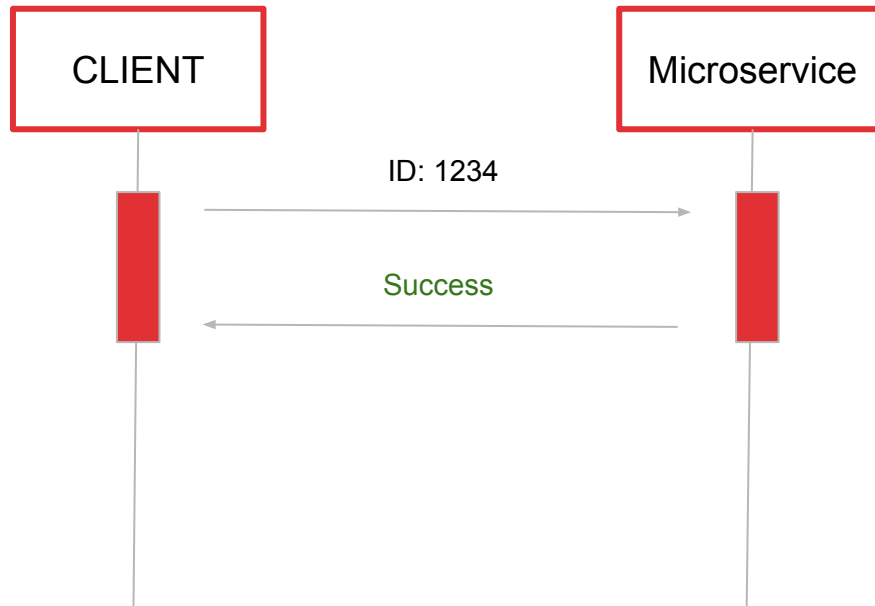
JSON Request

```
{  
  'request-id': `12345`  
  'email' : test@dialcom24.pl,  
  'name': 'Jan',  
  'surname': 'Kowalski',  
  'callbackUrl': 'https://example.pl/callbackUrl'  
}
```

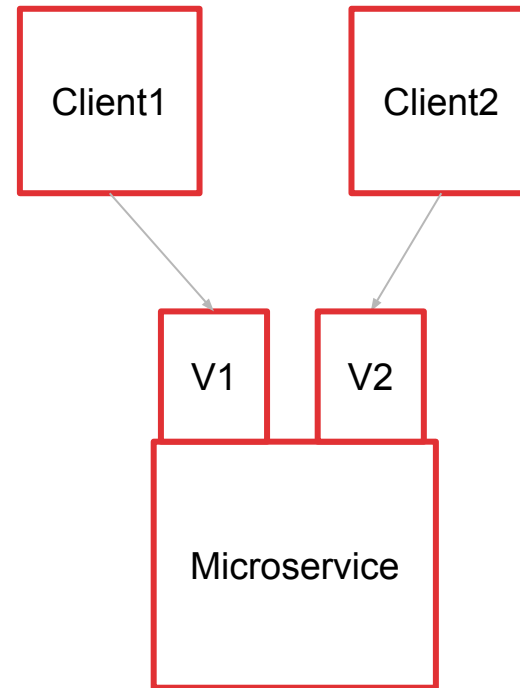
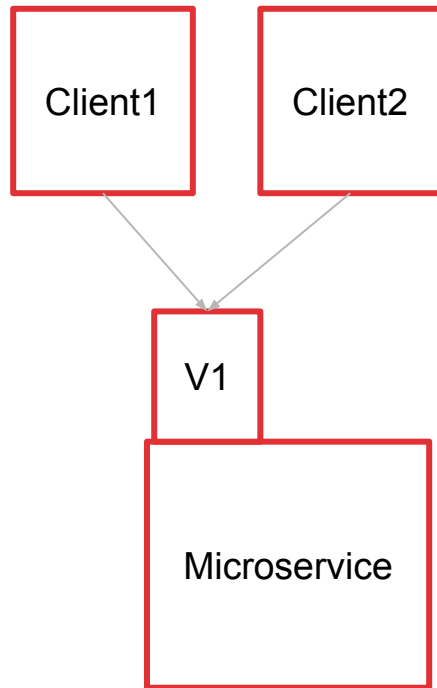
Idempotence

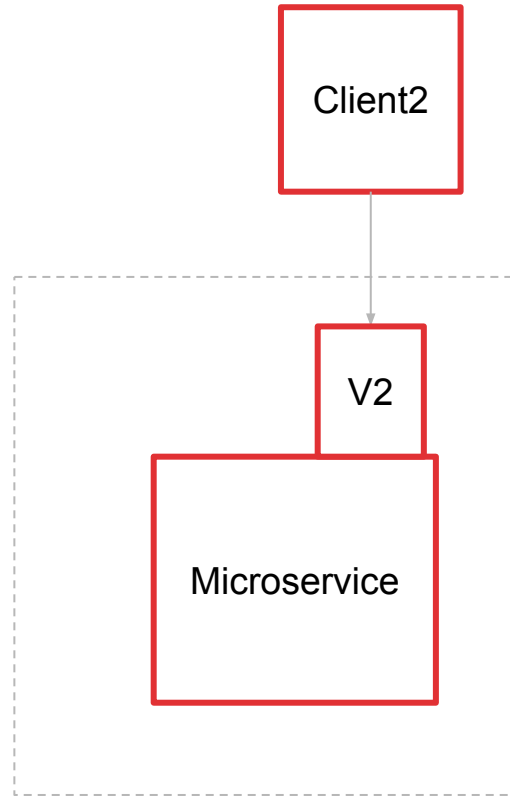
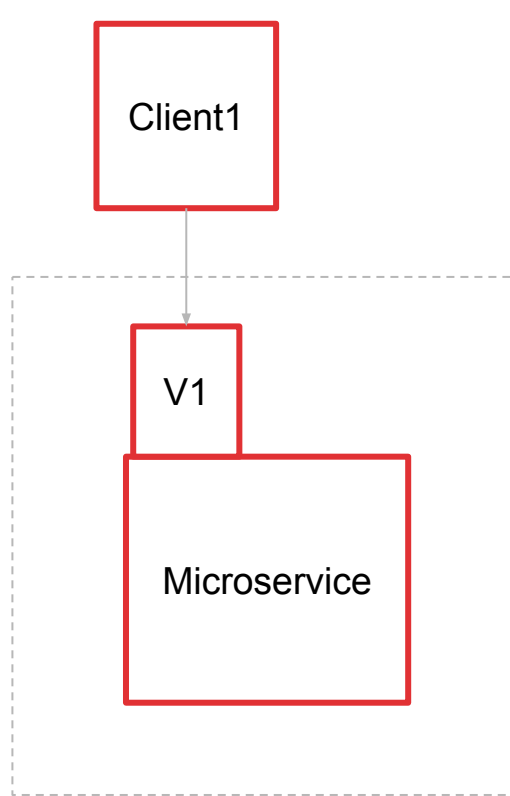


Idempotence



Versioning



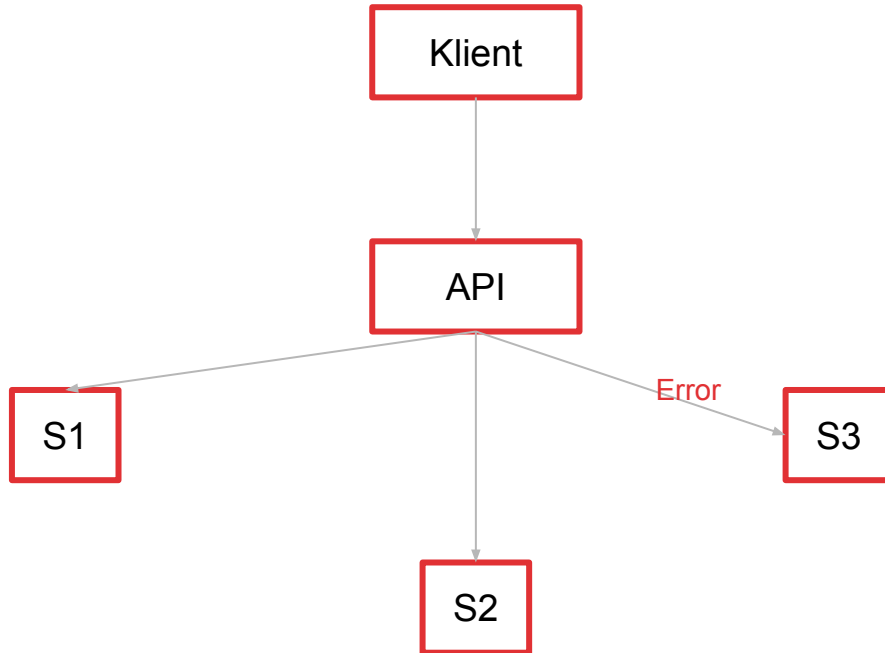


Semantic versioning

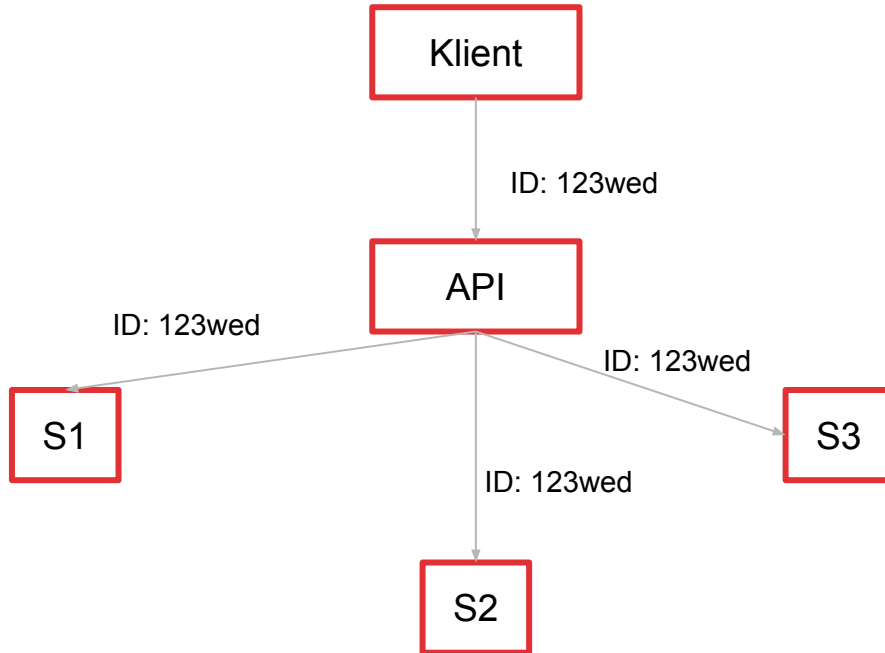
MAJOR.MINOR.PATCH

np. 2.0.0

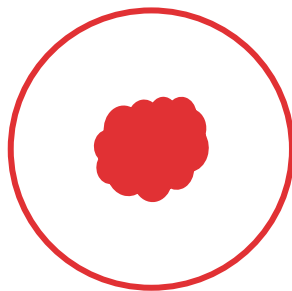
Monitoring



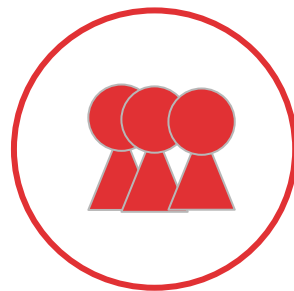
Monitoring



When to use?

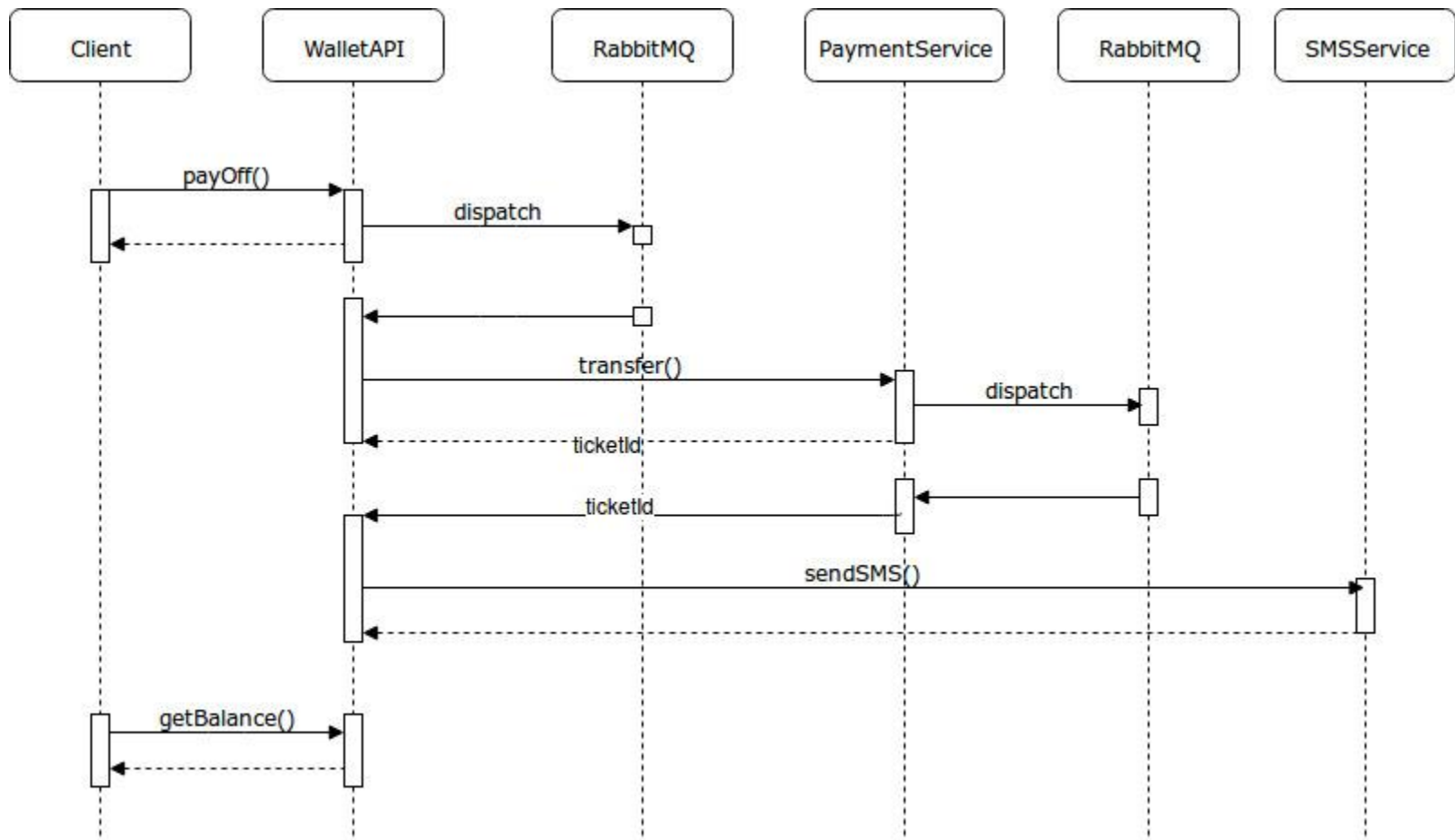


Resources



Teams

How to start?



Thank you

