

Microservices with PHP and ZeroMQ

...

Michael Osl

Before I start ...

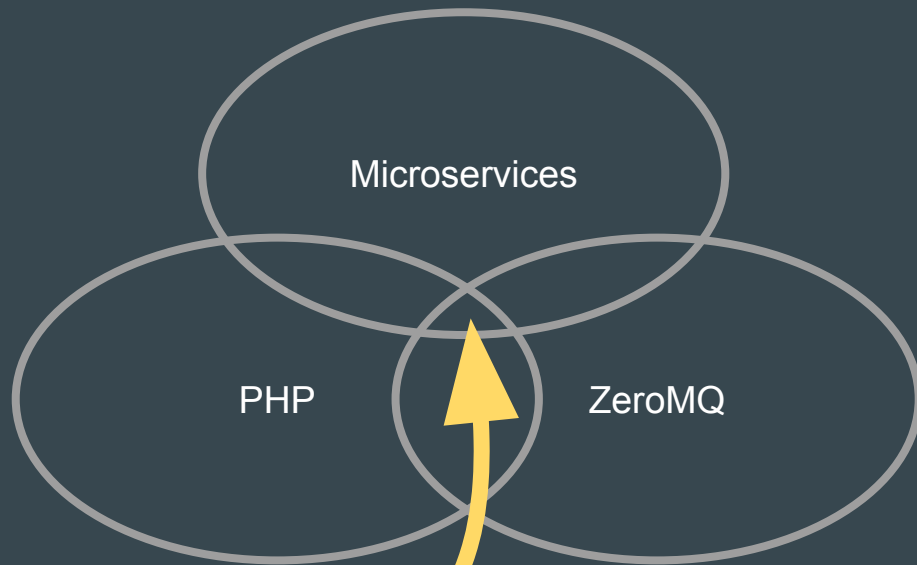
Notbremse

~~Handgriff nur bei Gefahr ziehen!~~

~~Jeder Mißbrauch wird bestraft~~

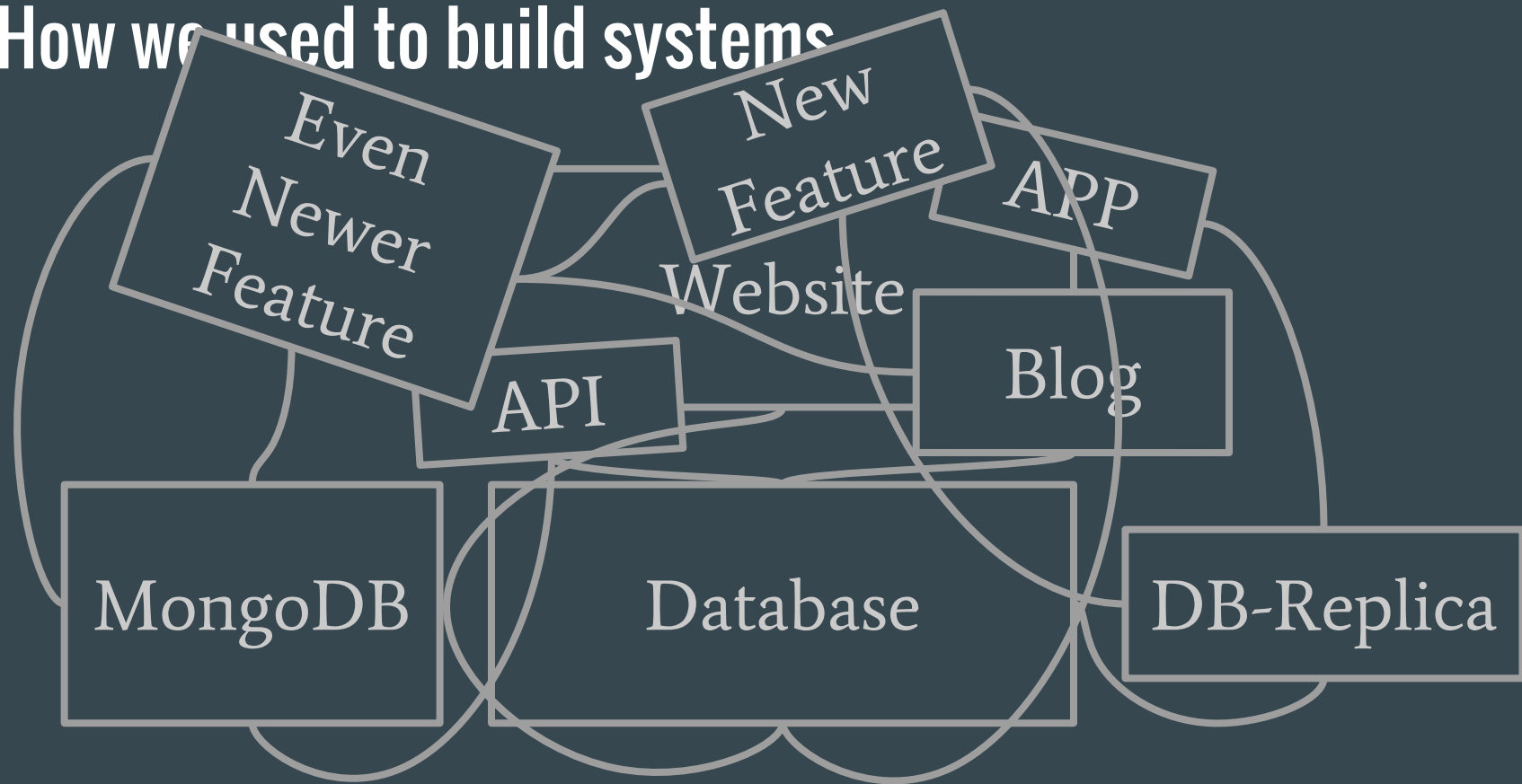
Agenda

- Microservices Intro
- ZeroMQ Intro
- ~~PHP Intro~~
- Bushtaxi
- Demo

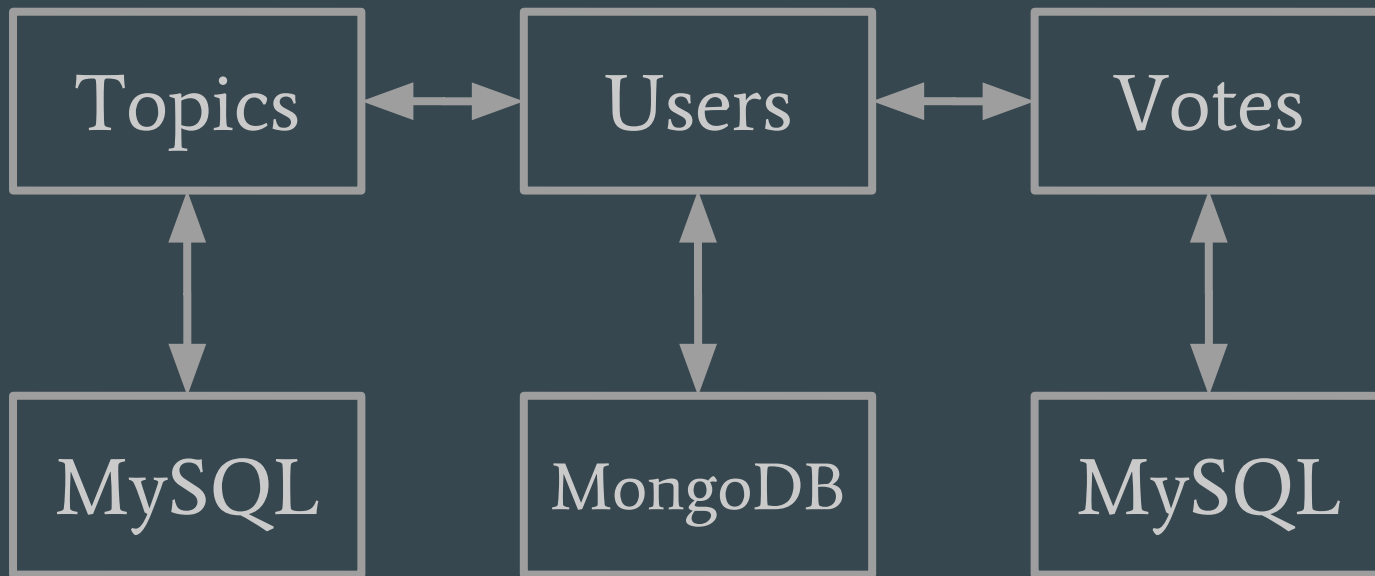


Microservices

How we used to build systems



How Microservices look like



Microservices on one slide

- Small, **independent** services
- Architectural style
- 12 Factor Apps
- “Micro”: Easy to understand
- Fostered by other developments

Deployment



```
graph LR; Deployment --> independent; Data --> independent; Development --> independent; Tests --> independent;
```

Data

Tests

Development

Can it make sense to use Microservices with PHP

YES

Service Integration

- How do services talk to each other?
- Synchronous
 - REST, RPC, ...
- Asynchronous
 - PUB/SUB

ZeroMQ

ZeroMQ

*“ZeroMQ (also known as ØMQ, 0MQ, or zmq) looks like an embeddable networking library but acts like a **concurrency framework**. It gives you sockets that carry atomic messages across various transports like in-process, inter-process, TCP, and multicast. You can connect sockets N-to-N with patterns like fan-out, pub-sub, task distribution, and request-reply. It's fast enough to be the fabric for clustered products. Its asynchronous I/O model gives you scalable multicore applications, built as asynchronous message-processing tasks. It has a score of language APIs and runs on most operating systems. ZeroMQ is from iMatix and is LGPLv3 open source.”*

-- <http://zguide.zeromq.org/page:all>

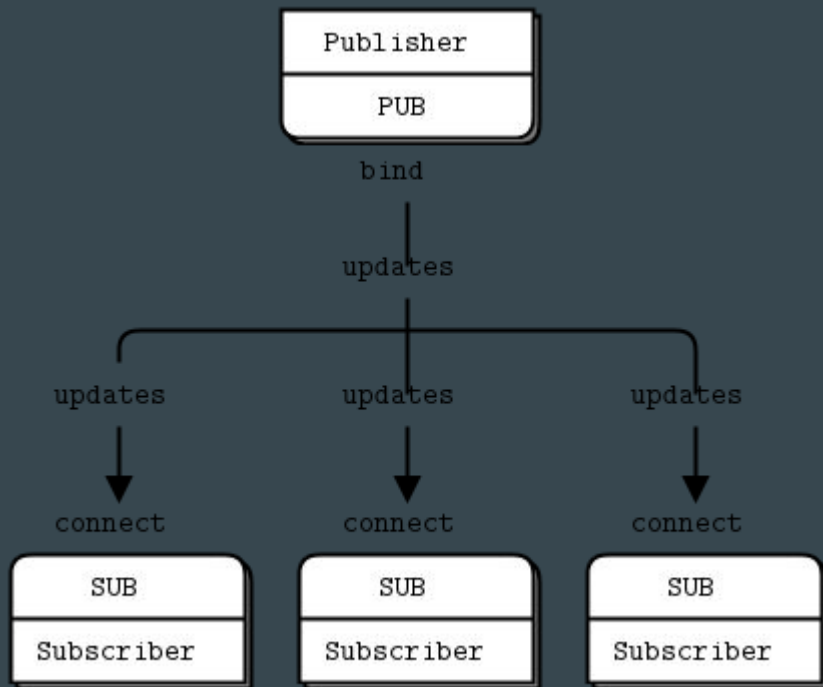
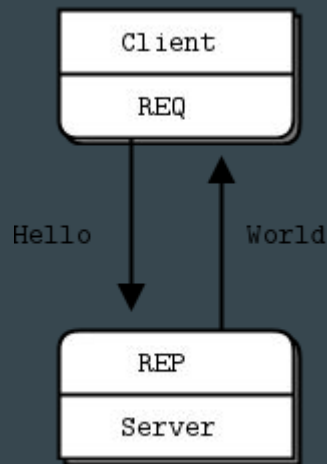
ZeroMQ

- It's **not** a messaging middleware
- It enables services to talk directly to each other
- Think of flexible sockets that enable various patterns (1:n, pub/sub, ...)
- Protocols: TCP, INPROC (and others)
- Handles the dirty stuff

ZeroMQ and PHP

- PECL Package available (only Beta version)
- Supports PHP 7
- Enables concurrency (which is not PHP's ...
ahemm ... core strength)

Messaging Patterns (Example)



Source: <http://zguide.zeromq.org/page:all>
License: cc-by-sa 3.0

Bushtaxi

Bushtaxi

- Composer package for easier handling of ZMQ
- PHP7 native
- Current state: Prototype
- <https://github.com/moe/bushtaxi>
- `composer require moe/bushtaxi`

Configuration (Client)

```
$config = [  
    "service" => [ "name" => "client" ],  
    "links" => [  
        "server" => [  
            "type" => "req",  
            "connect" => "tcp://127.0.0.1:5000"  
        ]  
    ]  
];  
$bushtaxi = new Bushtaxi\Client($config);  
$bushtaxi->server->send('Hello');  
print($bushtaxi->server->recv());
```

Configuration (Server)

```
$config = [  
  "service" => [ "name" => "server" ],  
  "links" => [  
    "client" => [  
      "type" => "rep",  
      "bind" => "tcp://127.0.0.1:5000"  
    ]  
  ]  
];
```

Configuration (Server), Part 2

```
$runtime = new class extends Bushtaxi\AbstractServerRuntime {  
    function handle($links) {  
        $message = $links['client']->recv();  
        $links['client']->send("World");  
    }  
};  
$bushtaxi = new Bushtaxi\Server($config, $runtime);  
$bushtaxi->run();
```

Demo

Not the best Demo in the world Demo

