redhat.

FROM LEGACY TO CLOUD

Pimping a ten year old webapp for the cloud

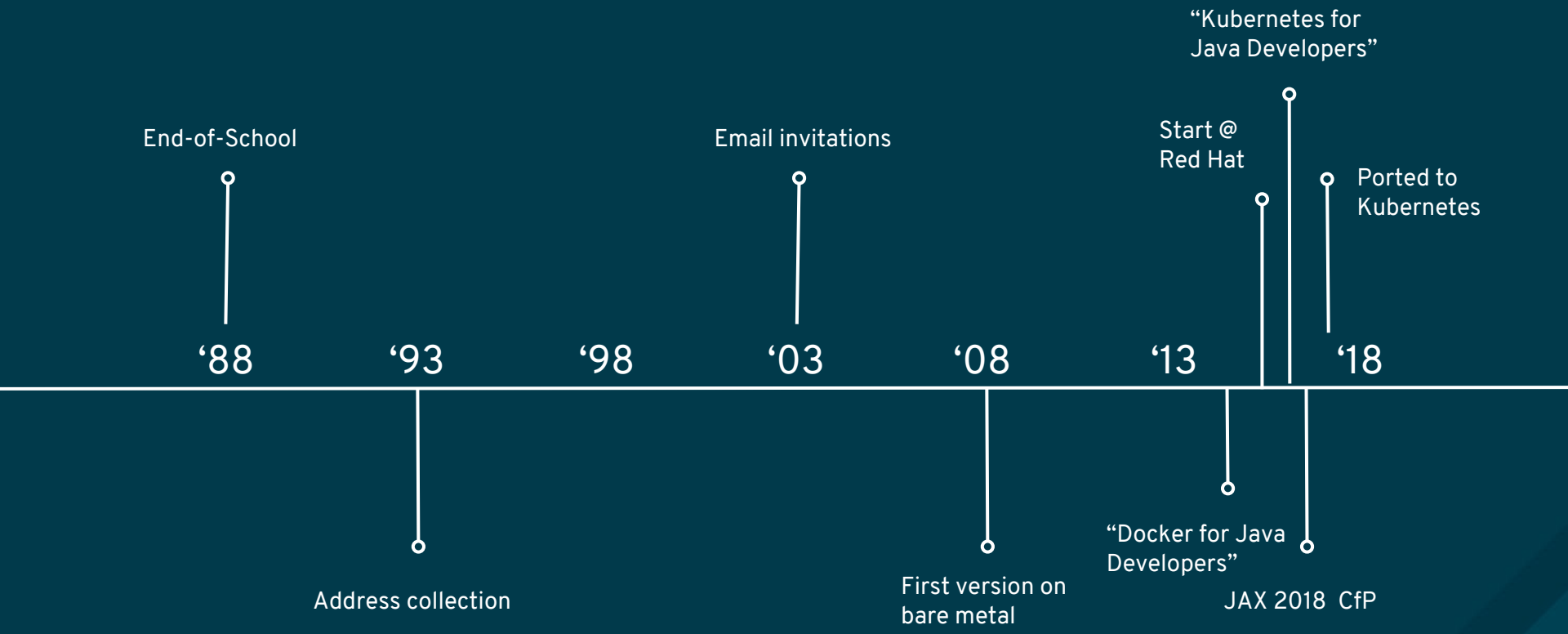
Roland Huss, Red Hat

JBCNConf - Barcelona - 2018-06-12

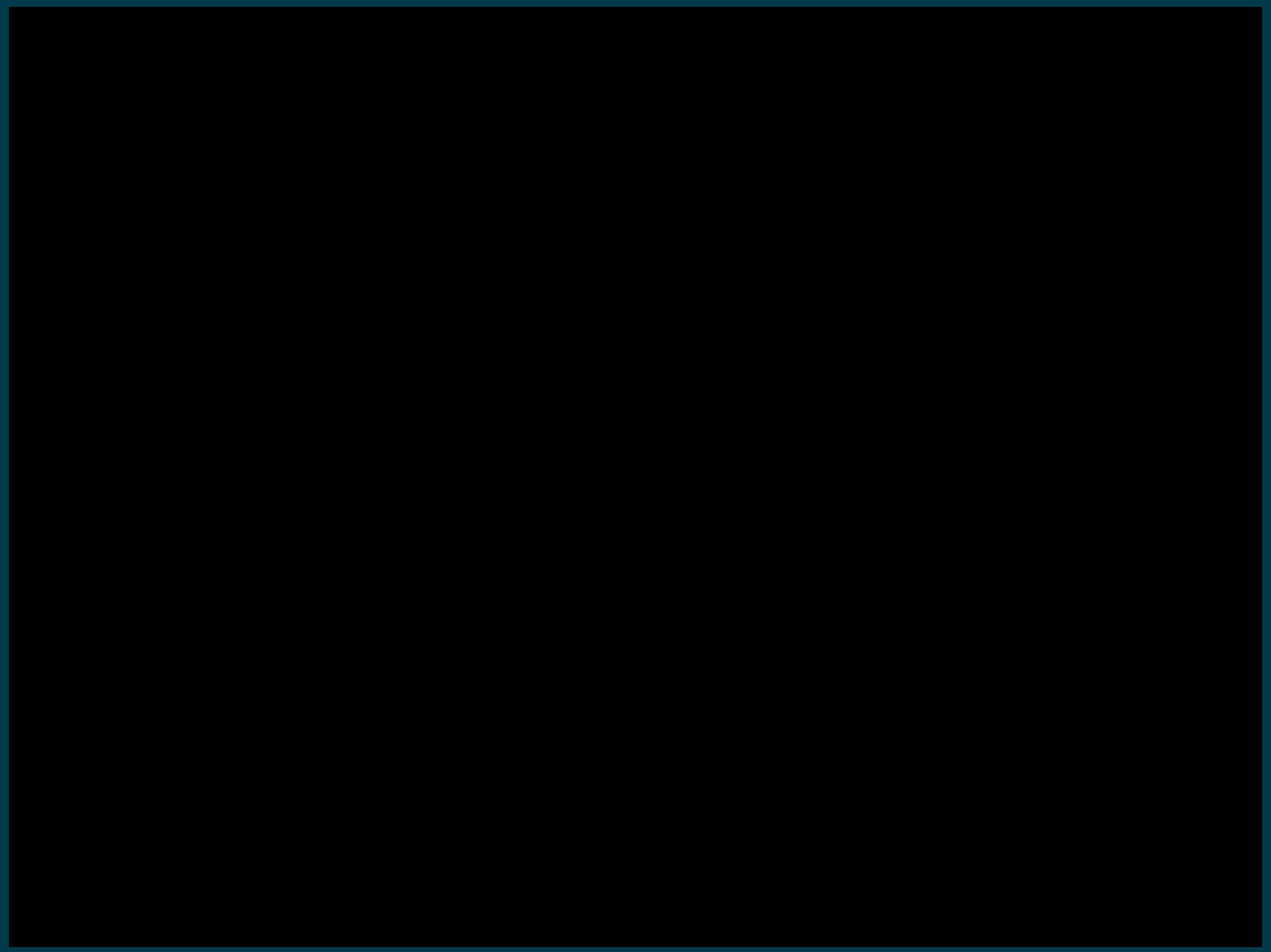
JUST AN EXAMPLE APP

ONCE UPON A TIME ...

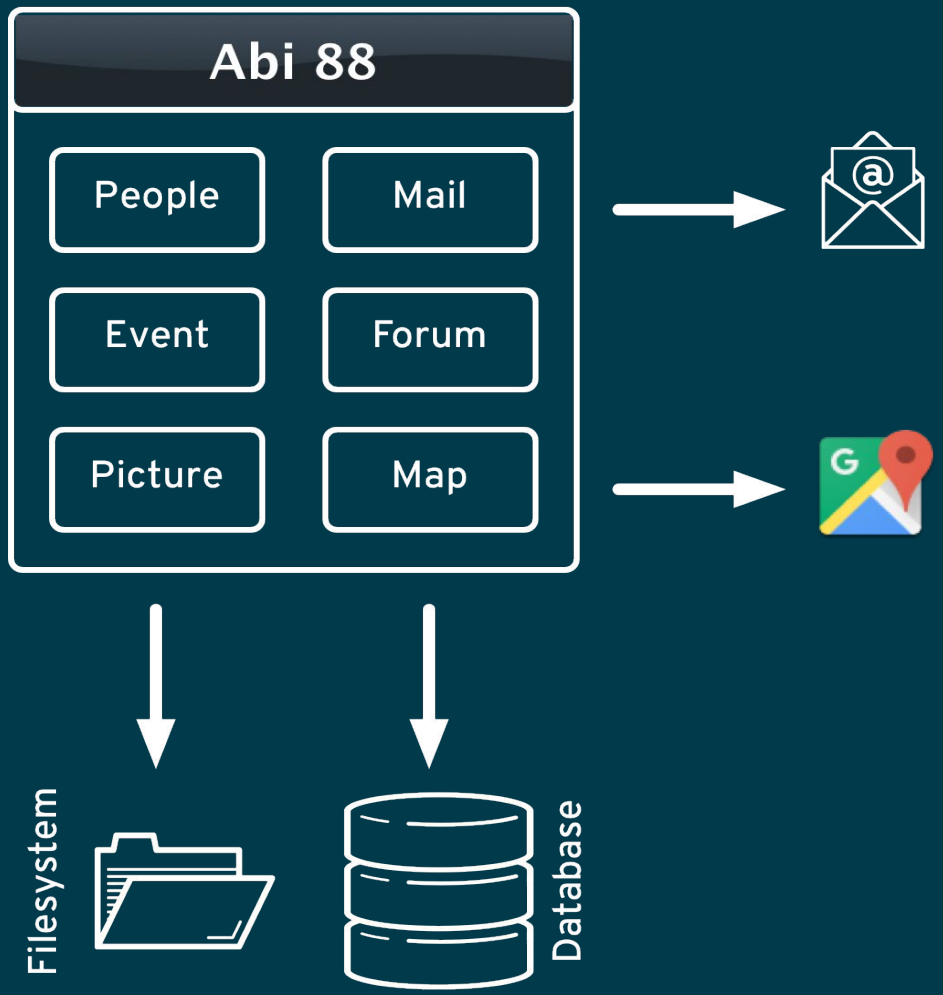
TIMELINE



WWW.ABI88-PEGNITZ.DE



ARCHITECTURE



TECH STACK 2008

Those were these days ...

Java 1.6

Wicket 1.4

Jetty 6

Spring 2.0

Hibernate 3.2

MIGRATION STRATEGIES

KEEP

Simple

Stack EOL

No security fixes

GMap component broken

Update later even more complicated

UPDATE

More work

Next updates easier

Latest Security patches

Better support on problems

Benefits of latest GMap improvements

DEMO

SPRING 2.0

```
2017-12-28 18:58:48,980 ERROR ontext.ContextLoader| Context
initialization failed
org.springframework.beans.factory.BeanDefinitionStoreException
: Unexpected exception parsing XML document from class path
resource [applicationContext.xml]; nested exception is
java.lang.IllegalStateException:
AnnotationTransactionAttributeSource is only available on Java
1.5 and higher
```

SPRING 2.0 JAVA VERSION DETECTION

```
static {
  javaVersion = System.getProperty("java.version");
  // version String should look like "1.4.2_10"
  if (javaVersion.indexOf("1.7.") != -1) {
    majorJavaVersion = JAVA_17;
  }
  else if (javaVersion.indexOf("1.6.") != -1) {
    majorJavaVersion = JAVA_16;
  }
  else if (javaVersion.indexOf("1.5.") != -1) {
    majorJavaVersion = JAVA_15;
  }
  else if (javaVersion.indexOf("1.4.") != -1) {
    majorJavaVersion = JAVA_14;
  }
  else {
    // else leave 1.3 as default (it's either 1.3 or unknown)
    majorJavaVersion = JAVA_13;
  }
}
```

MIGRATION

2008	2018
Java 6	Java 8
Spring 2.0	Spring 3.2
GMap API 2	GMap API 3
Wicket 1.4	Wicket 6.0 (via 1.5)
Jetty 6	Jetty 9
MySQL 5.0	MySQL 5.5

CHALLENGES

- Spring 3.2:
  - biggest change: Switch from single Spring dependency to many smaller deps
- Wicket 6.0
  - Wicket 1.4 -> 1.5 : 6 hours
  - Wicket 1.5 -> 6.0: 30 mins
  - Some components have been vapourized
  - Don’t try to migrate to generics. Takes ages with no real benefit
  - GMap2 to GMap3 migration was a bit complicated, too
- Jetty 9, MySQL 5.5: No problems at all
- Hibernate **not** updated

HOSTSELECTOR - XML

```
<bean id="mailService"
    class="org.abi88pegnitz.service.impl.MailServiceImpl">
  <!-- .... -->
  <property name="hostSelector">
    <bean class="org.abi88pegnitz.util.HostSelector">
      <property name="mapping">
        <map>
          <entry key="localhost" value-ref="homeMailConfig"/>
          <entry key="194.246.123.70" value-ref="peopleMailConfig"/>
          <entry key="194.246.123.67" value-ref="peopleMailConfig"/>
          <entry key="94.185.90.70" value-ref="peopleMailConfig"/>
        </map>
      </property>
    </bean>
  </property>
</bean>
```

HOSTSELECTOR - CODE

```
public class HostSelector {

  // Key: IP/Hostname, valu: config Object
  private Map mapping = new HashMap();

  public Object select() {
    String requestServerName = getServerNameFromRequest();
    InetAddress addr = getInetAddress();

    // Try IP first
    for (String key : new String[] {
      addr.getHostAddress(), requestServerName,
      addr.getHostName(), addr.getHostAddress(),
      addr.getCanonicalHostName(), "default" }) {

      if (key != null && mapping.containsKey(key)) {
        return mapping.get(key);
      }
    }
    return null;
  }
}
```

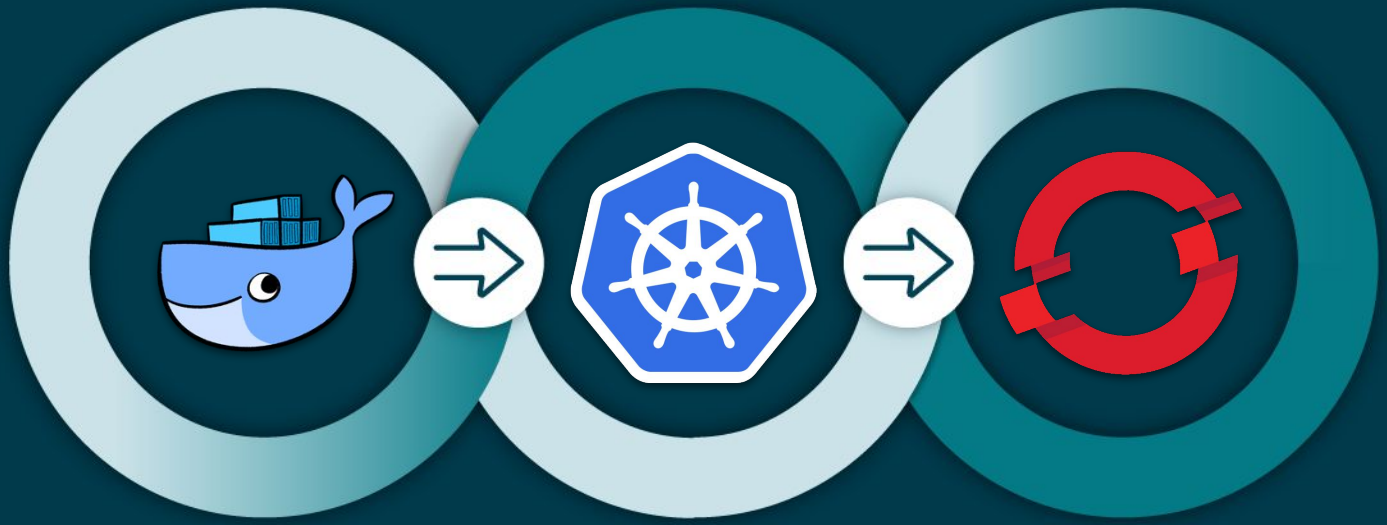
CONFIGURATION

- “HostSelector”: Bad idea, as IPs are highly volatile in a Cloud Environment
- Use environment variables instead

- `application.properties` :

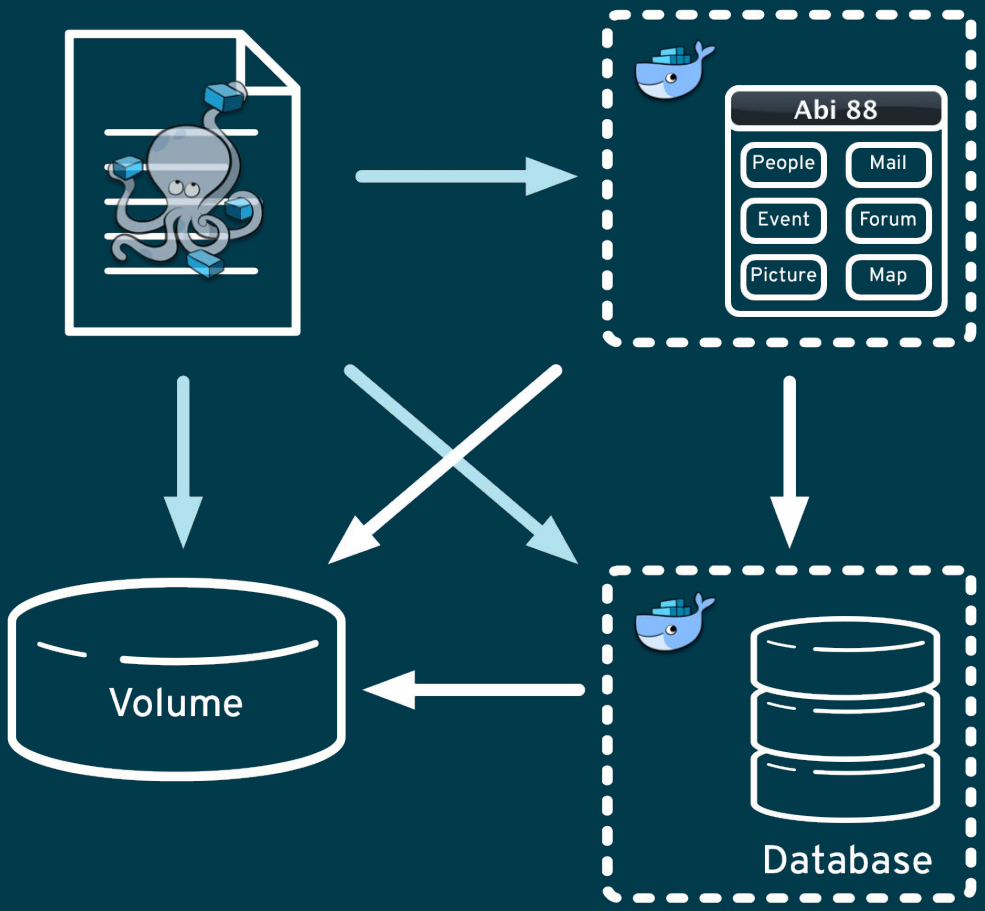
```
mail.smtp.host      =${MAIL_SMTP_HOST}
mail.smtp.user      =${MAIL_SMTP_USER}
mail.smtp.password  =${MAIL_SMTP_PASSWORD}
mail.smtp.debug     =${MAIL_SMTP_DEBUG}
mail.from.address   =${MAIL_FROM_ADDRESS}
```

ROADMAP



“Containerize” “Orchestrate” “Enterprize”

CONTAINERIZE



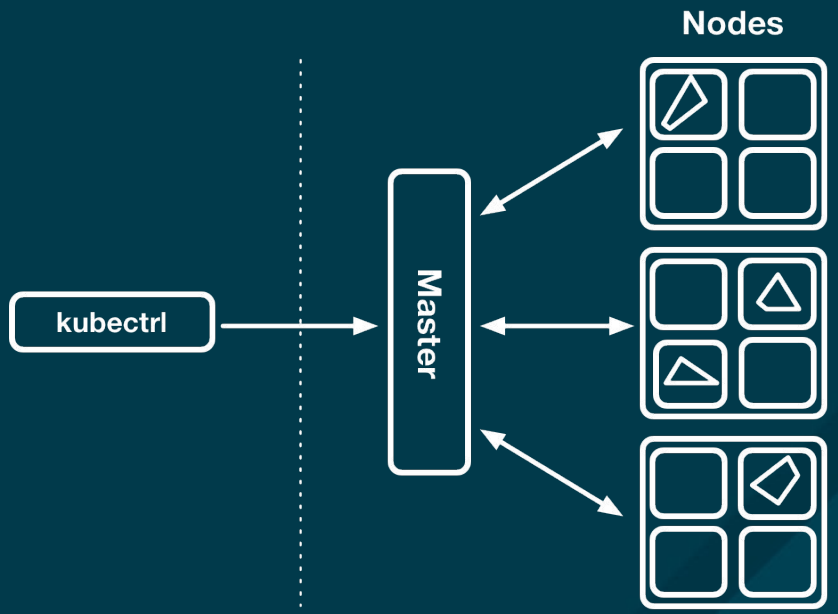
DEMO

DOCKER-MAVEN-PLUGIN

- Last supported docker-maven-plugin
- Features:
  - **Building** images & **Running** containers
  - Dockerfiles and Docker compose files
  - Multiple image configurations
  - Extensive authentication support also for external cloud providers (ECR, Google Container Registry)
  - Advanced Docker features like custom networks or volumes
- Documentation: https://dmp.fabric8.io

KUBERNETES

- Open Source project started 2014 by Google engineers
- Orchestration system for containers
  - Scheduling
  - Horizontal scaling
  - Self-healing
  - Service discovery
  - Rollout / Rollback
- Declarative resource based API
- Current version: 1.10



HOSTING OPTIONS

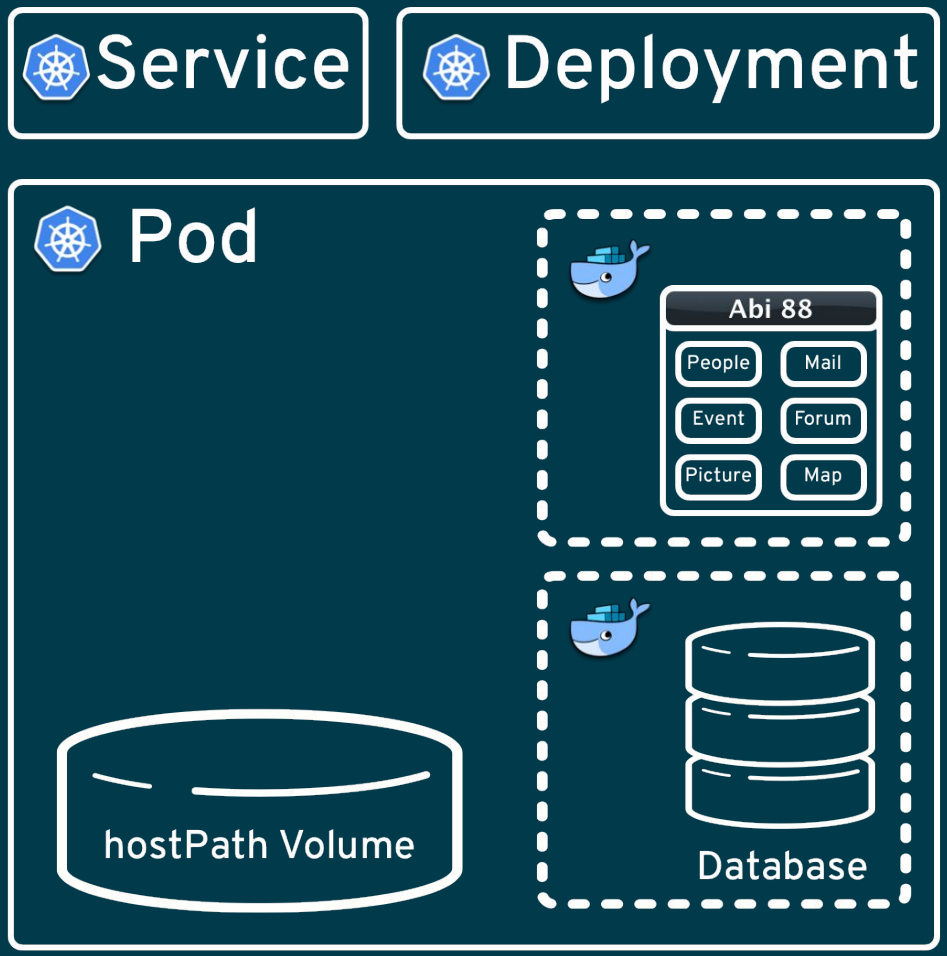
- CaaS (1 Node):
  - Google Kubernetes Engine: 14 EUR / month
  - OpenShift Online Pro: 40 EUR / month
  - Azure Container Service: 13 EUR / month
  - OpenShift Online Starter: free
- IaaS (1 Node, 2GB):
  - Digital Ocean: 8 EUR / month
  - Scaleway: 3 EUR / month
  - ...

https://www.webstack.de/blog/e/cloud-hosting-provider-comparison-2017/

SCALEWAY

- 2 GB / 2 Cores / 50 GB SSD / 1 public IP
- Installation with **kubeadm**
- **Træfik** as Ingress Controller
- Let's encrypt support
- Own Domain name
- Mailgun as SMTP service, but then switched back to GoogleMail
- Monitoring via Nagios from remote
- Costs: **~ 200** EUR for five years

SIMPLE ORCHESTRATION



DEMO

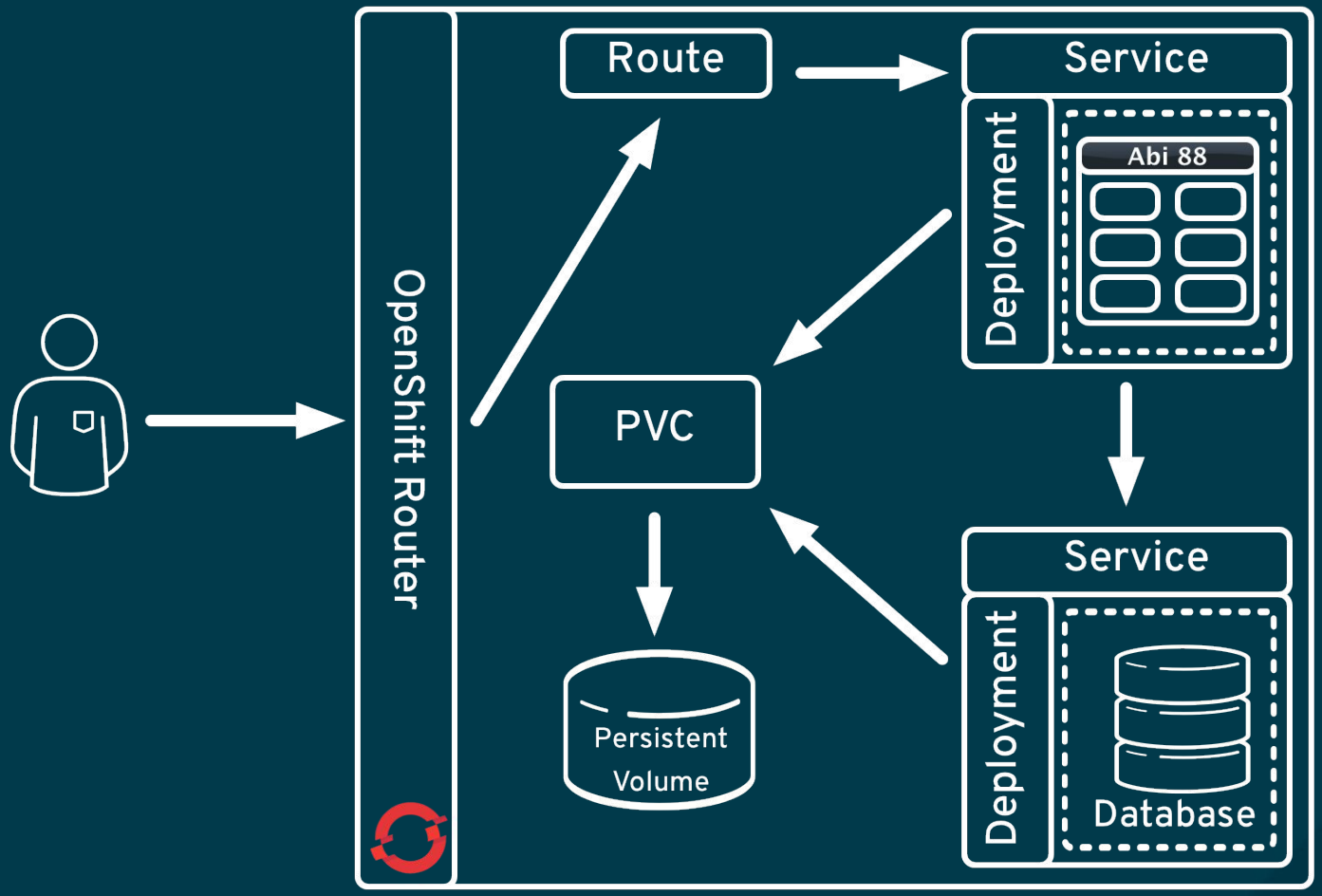
FABRIC8-MAVEN-PLUGIN

- Opinionated Maven Plugin for creating Kubernetes and OpenShift resource descriptors
- Focus on openshift.io support but also mostly general purpose
- 3 configuration modes:
  - **Zero Config** with highly opinionated defaults
  - **XML** Plugin configuration
  - **Resource fragments** which gets enriched for the missing parts
- Support for debugging and hot redeployments

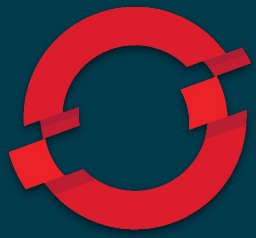
SINGLE POD IS TOO SIMPLISTIC

- Coupled Lifecycle of Application and Database
- Not scalable at all
- `hostPath` Volume: Only for Clusters with 1 Node useful

PROPER ORCHESTRATION



OPENSHIFT



- Adds the **BUILD** aspect to Kubernetes
  - “Source-to-Image” Technology
  - Link between Deployment and Build allows automatic rebuild and redeployments
- Infrastructure Services
  - Registry
  - Router
  - OAuth2 SSO
  - Multi tenancy
  - Management UI
  - Ansible Service Broker
  - ....

OPENSHIFT ONLINE STARTER

Metric	Limit
Memory	1 Gi
Persistent Volumes	1
Storage	1 Gi
Minimum Memory Request	100 Mi
Limit / Request Ratio	50%

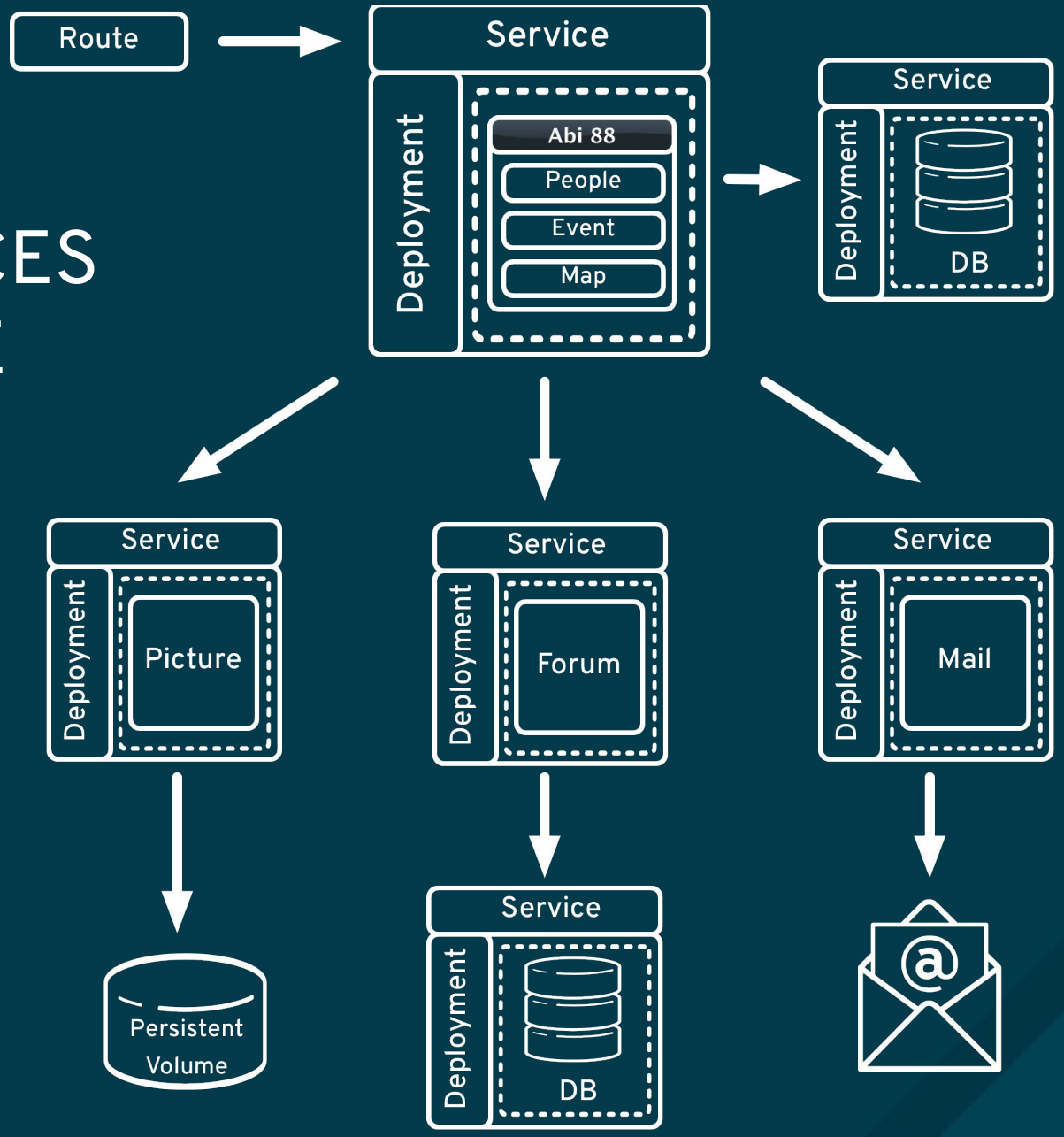
OPENSHIFT TEMPLATES

- Deployment descriptor for whole applications
- Simple to use
- Similar to Helm charts, but more limited
- Many predefined templates on OpenShift
- New: Ansible Playbook Bundles as an alternative to Templates

DEMO

WHAT'S NEXT

MICROSERVICES EVERYWHERE



OTHER IDEAS

- Introduce a **Service Mesh** like Istio for enhanced resilience and tracing
- Evaluate a **MySQL cluster** on Kubernetes (Galera ?)
- Monitoring with **Prometheus**
- Introduce session affinity for ingress controller to horizontally scale Web app
- Look how to **decompose UI** componentwise (e.g like the SCS approach)

WRAP UP

CODE MIGRATION WAS THE HARDEST PART

Some many dragons are waiting for you

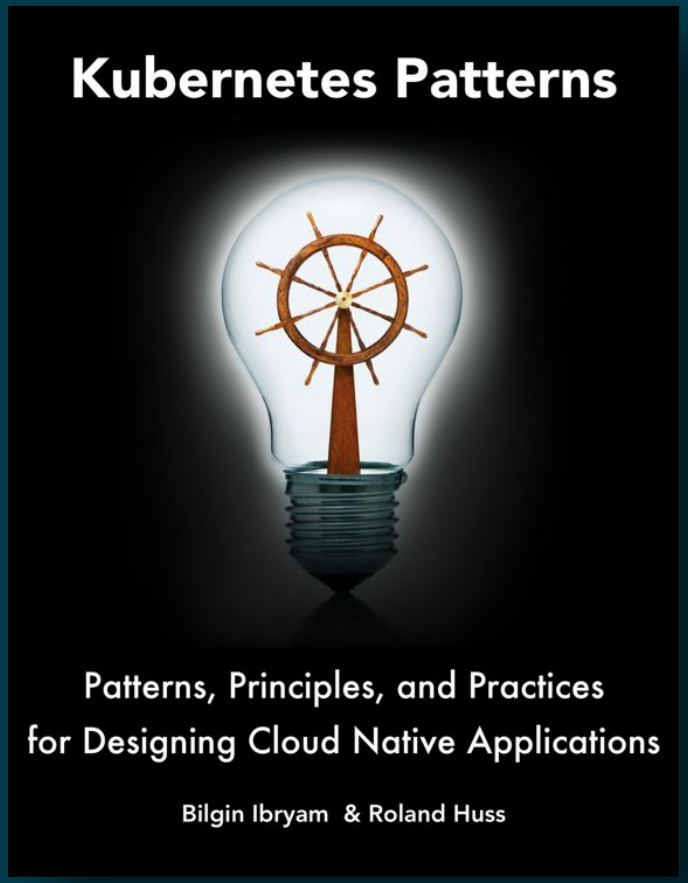
TOOLING HELPS IN JUMPING ONTO THE CLOUD

But tooling support can still be improved

KUBERNETES AND OPENSHIFT HAVE GROWN UP

Both are awesome orchestration platforms, rock stable and still accelerating

EVEN MORE KUBERNETES GOODNESS



https://leanpub.com/k8spatterns

redhat.

THANK YOU

@ro14nd