

# Building Modular Java Applications for the Cloud



amptu



# About me



Bert Ertman

Fellow at Luminis (Netherlands)  
JUG Leader for NLJUG  
Java Champion

**luminis** 

[bert.ertman@luminis.eu](mailto:bert.ertman@luminis.eu)



@BertErtman





## Observations

- Extremely agile  
(first mover advantage)
- Architecture (and code base)  
should be able to cope with change

## Trend

- Applications tend to  
grow bigger and more  
complex
- Agile development  
and refactoring have  
become more common



This leads to a number of challenges :

Dependency  
management

Versioning


Maintenance  
(long term)

Deployment









Modularity  
is the answer



## The case for modularity

Modularity is the ultimate agile tool!

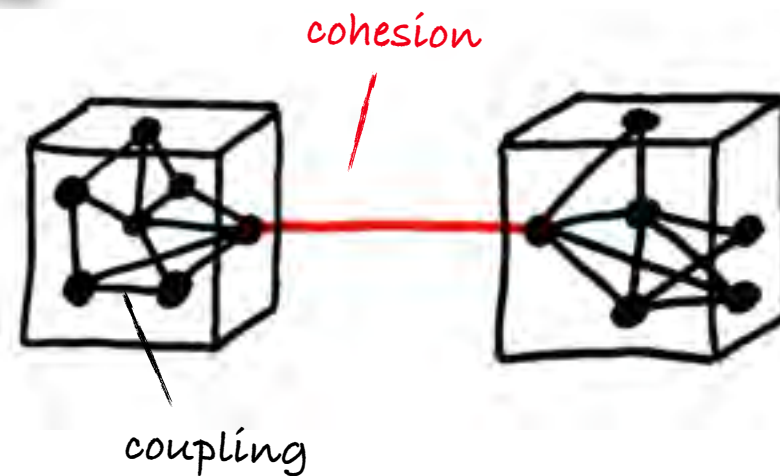
- Small, disposable, components
- Prevents code rot on the architectural level
- Isolate problems, focus work




# What we learned about OO design in university :

Prevent  
(tight)  
coupling

Promote  
cohesion








How to prevent  
coupling in a code  
base?






How to prevent  
coupling in a code  
base?

interfaces





But how to make  
sure nobody  
accidentally uses  
implementation  
classes?





imp

lemen

tation

hi

di

ng

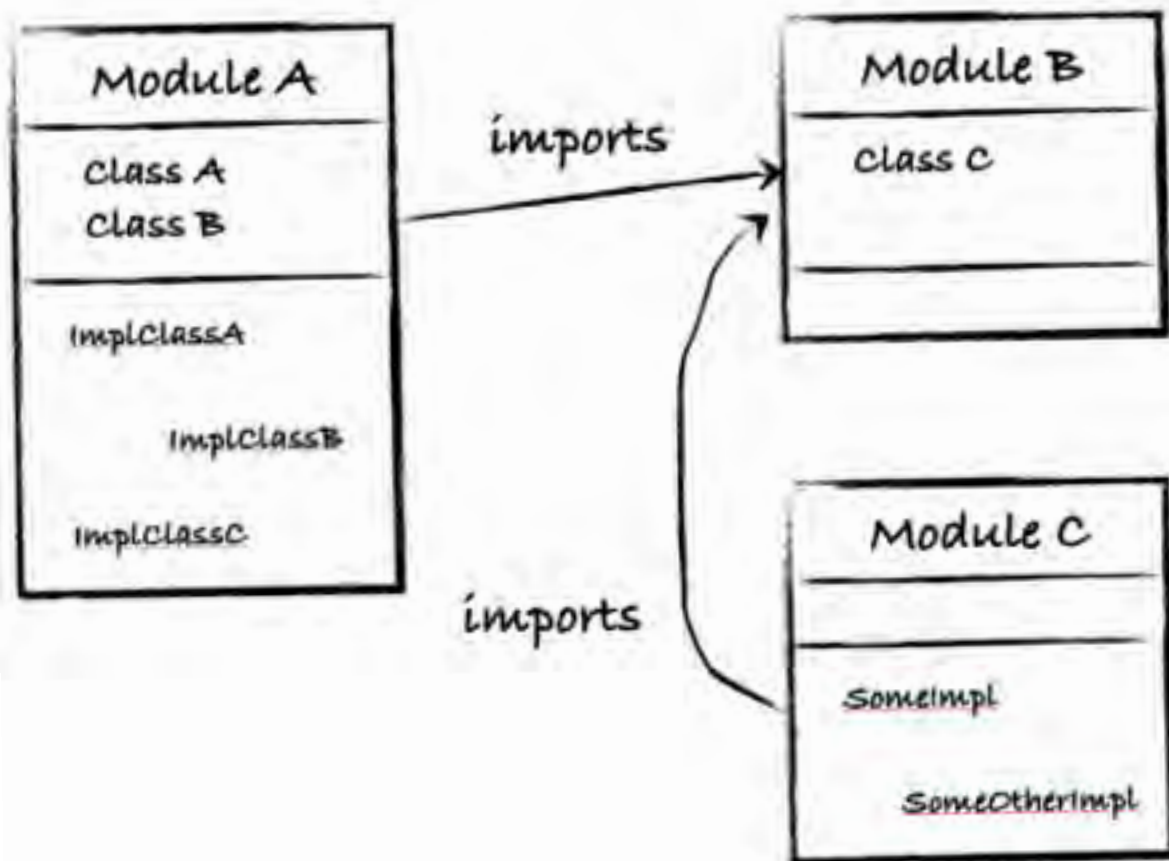


Ok, but how to create an instance of a hidden class?

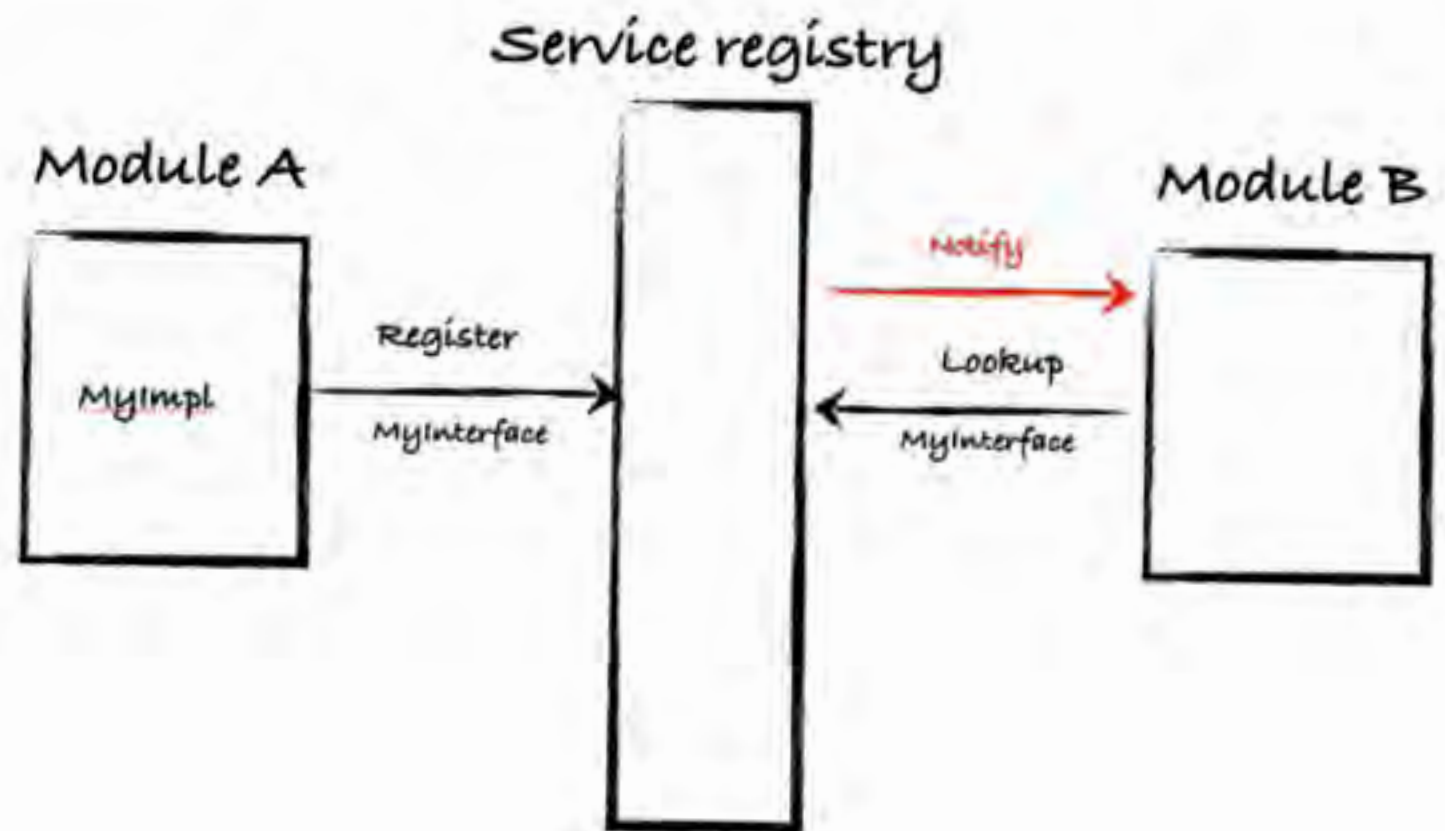
```
MyInterface myI = new  
MyImplementation();
```



# Modules



# Service Lookups







Stop talking...  
and show me code!





Back to the cloud...

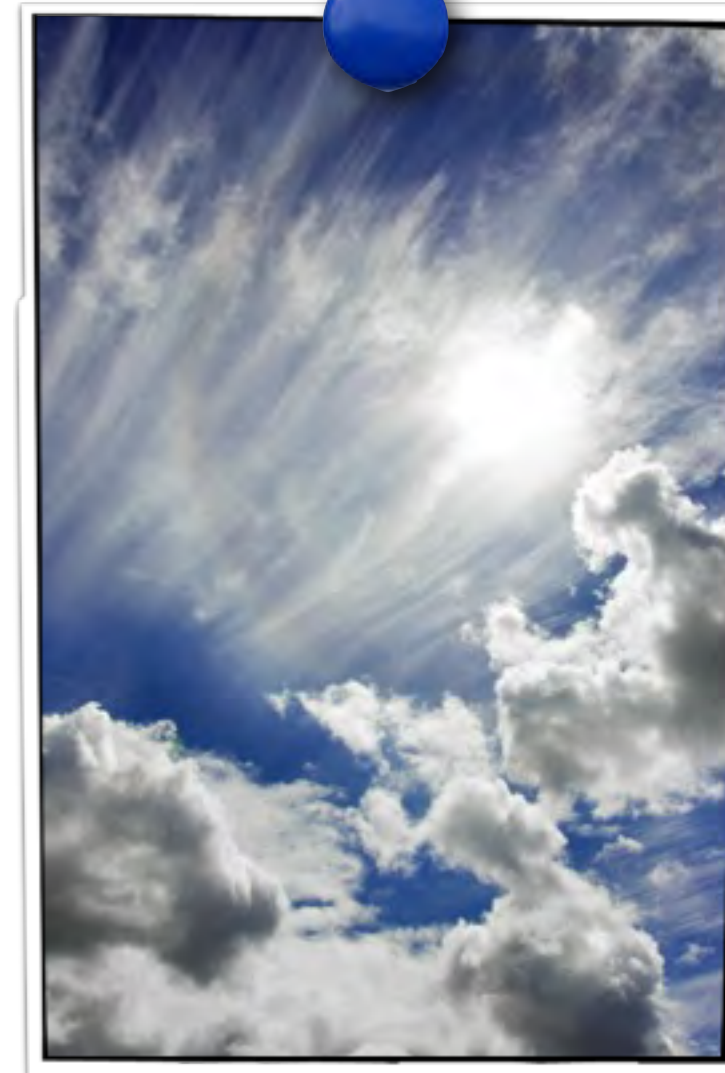


# Apps are moving to the cloud

Application as a Service over the Internet

Cloud challenges require non-trivial non-functional requirements:

- Zero-downtime deployments
- Modular deployments
- Customer specific extensions (SaaS)





# Requirements:

- ❑ Modern web app
- ❑ UI mostly offloaded to clients or devices
- ❑ Document driven interaction
- ❑ Integration via REST API
- ❑ Web scale data store
- ❑ Multi-tenant
- ❑ Elasticity

# Typical architecture

HTML 5 + JavaScript

RESTful services

OSGi services

Apache Felix

Mongo

S3



# Components

- Auth
- Blob stores
- MongoDB
- Multi-tenancy
- OpenSocial
- Search
- Remote Services
- REST
- Template
- Web
- ...



**amdatu**



Let's Add **AMDATU**  
to our stack →

- Architectural focus on modularity
- Runtime dynamic services
- High level API

# Typical architecture

HTML 5 + JavaScript

A  
m  
d  
a  
t  
u

RESTful services

OSGi services

Apache Felix


Mongo

S3



Demo

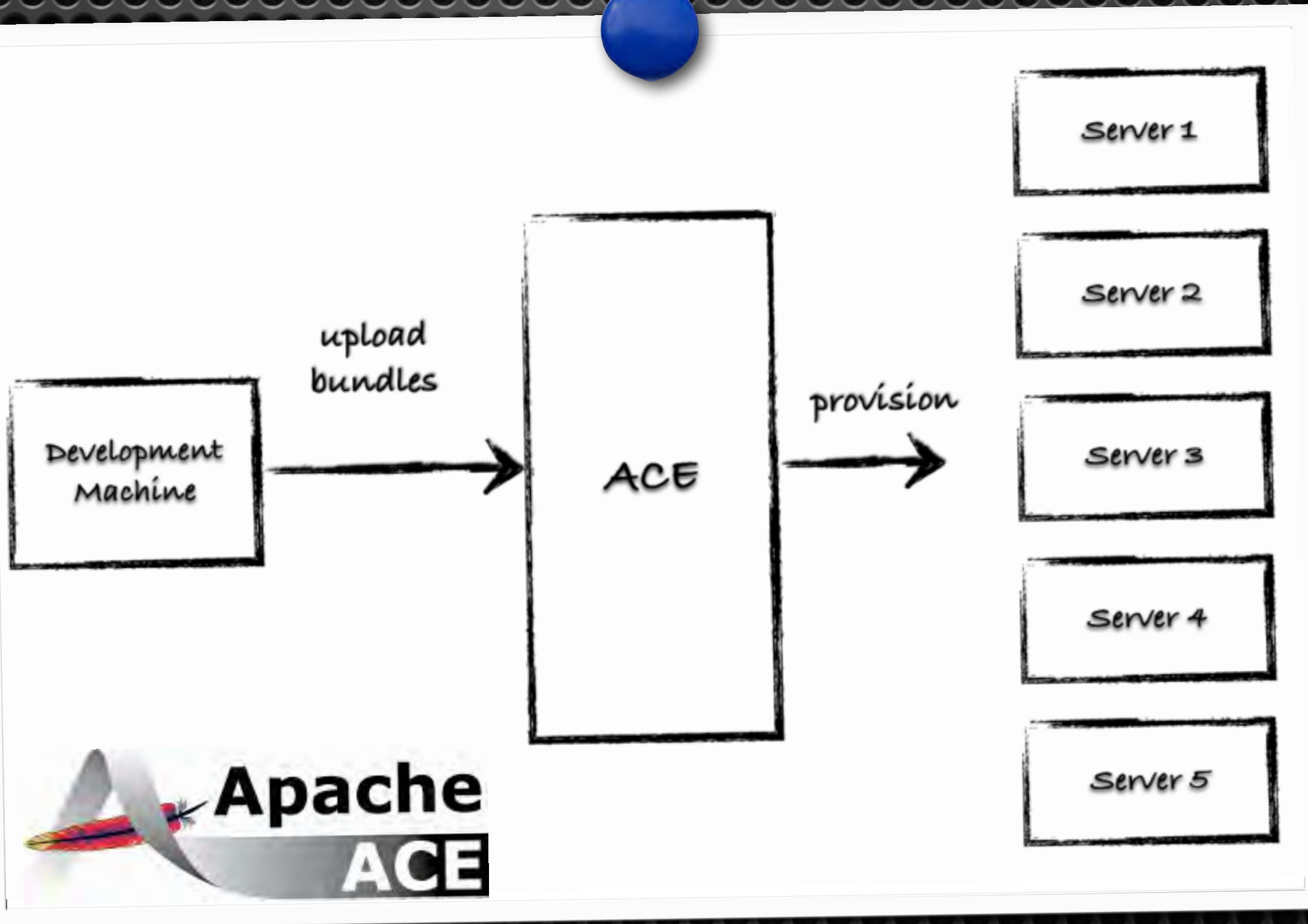




What about  
deployment?



# Provisioning Server





Demo

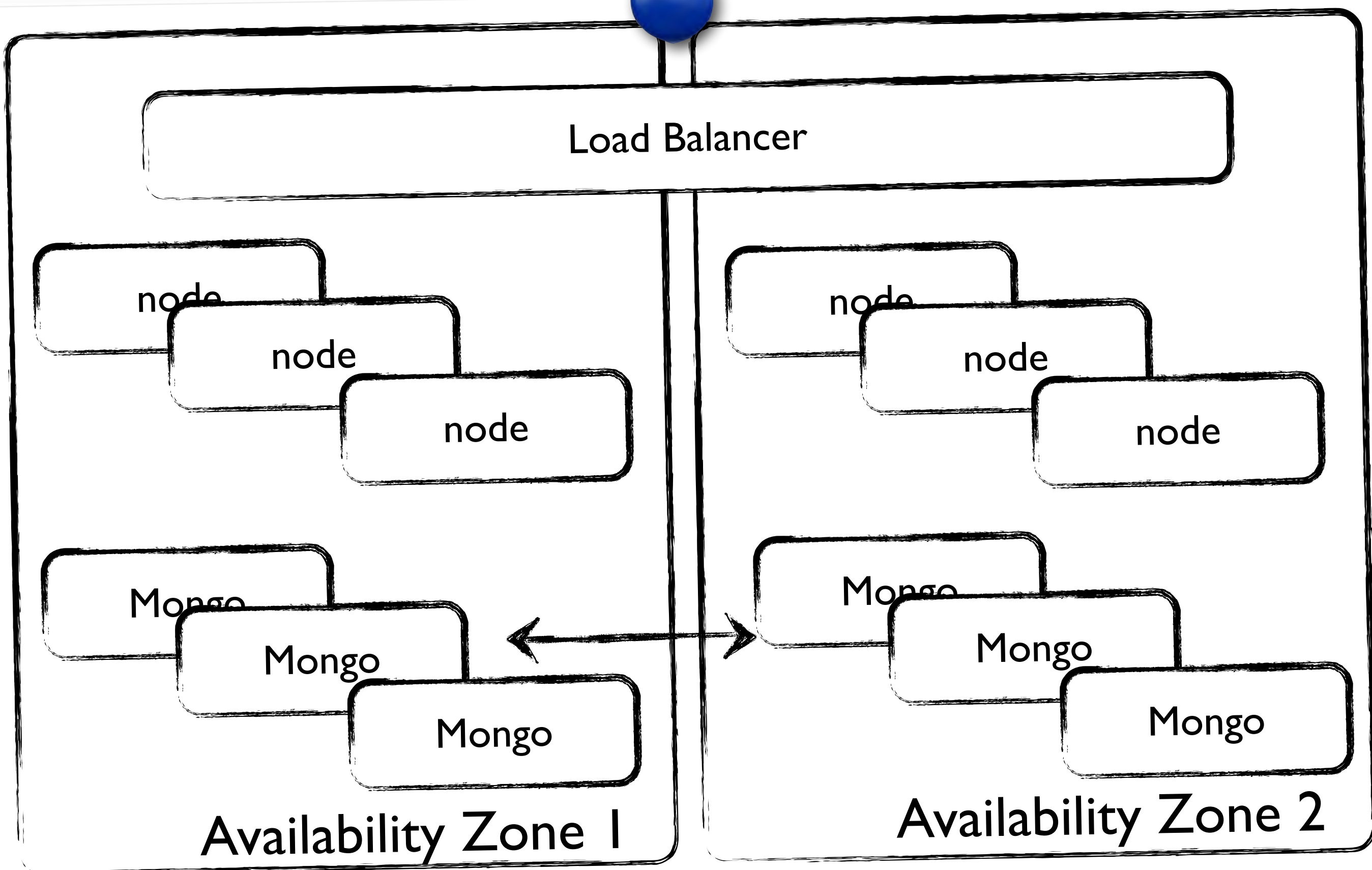


# Auto scaling

Enough  
capacity

Without paying  
for idle servers  
at night...





Load Balancer

node

node

node

Mongo

Mongo

Mongo

Availability Zone 1

node

node

node

Mongo

Mongo

Mongo

Availability Zone 2



# Auto scaling

AWS Auto Scaling

1. Start

Node

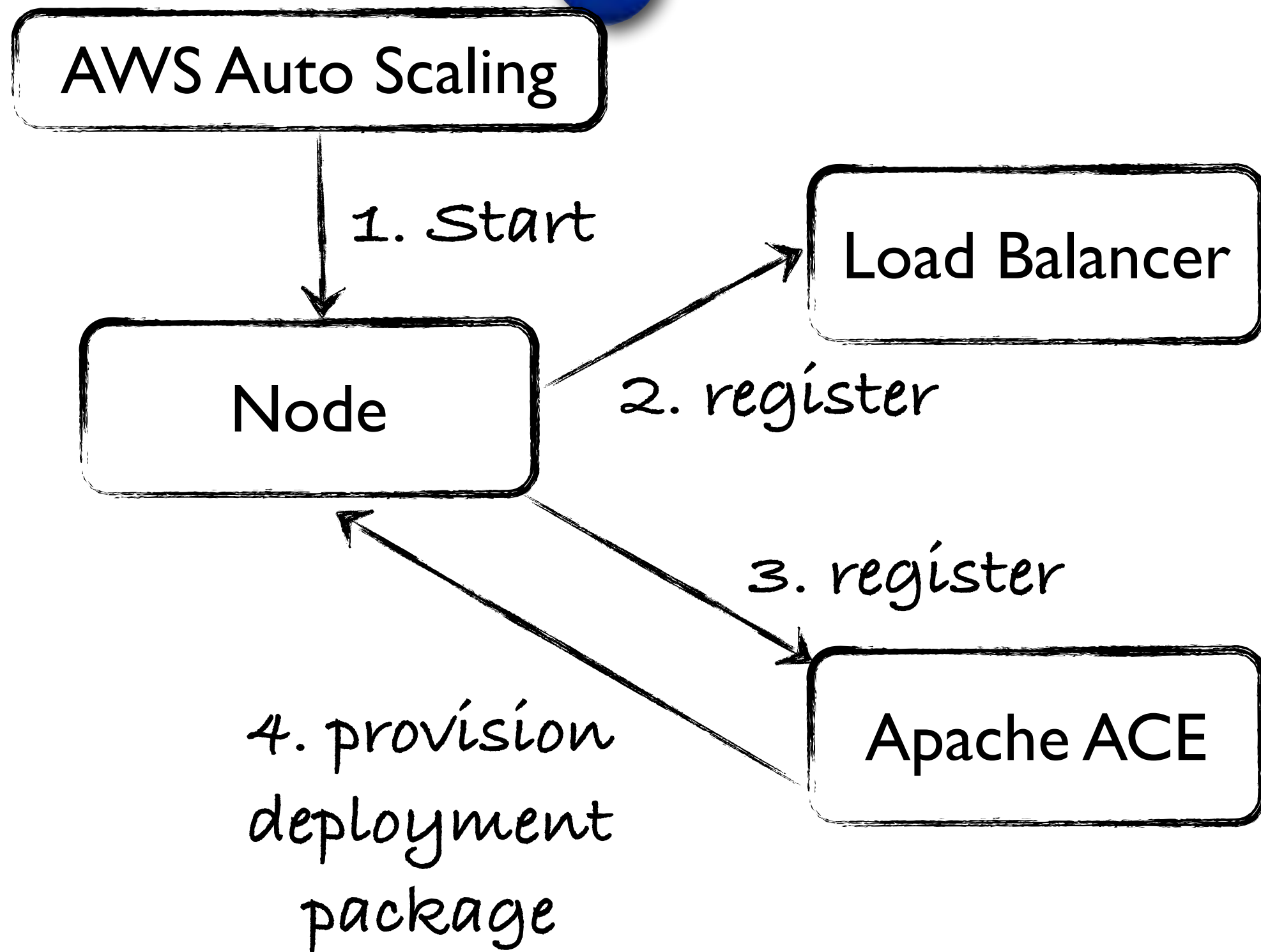
Load Balancer

2. register

3. register

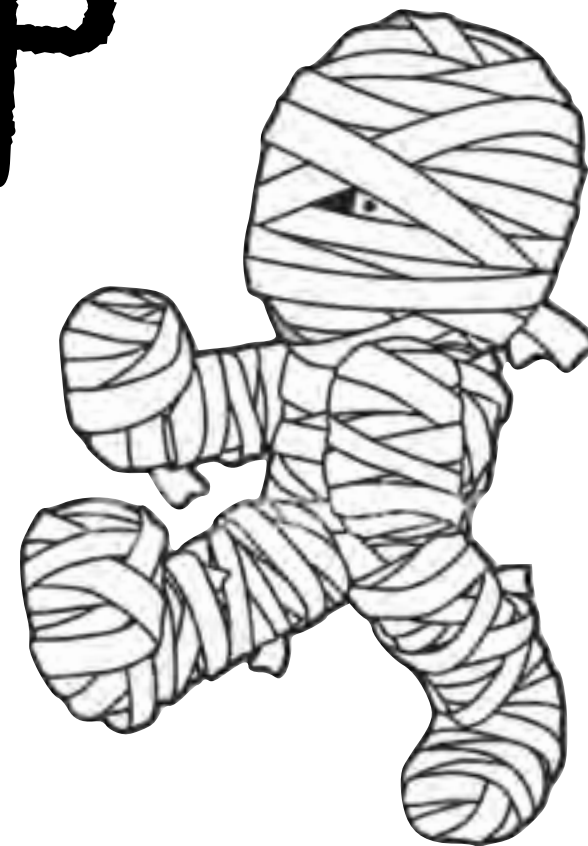
Apache ACE

4. provision  
deployment  
package





wrap up







## What have we learned?

- Why modularity is important (in the cloud)
- Practical solution for doing modularity now
  - using an Open Source development stack
  - using ready to use Cloud Components
- We are using this for high-profile production applications!



Practical Modular Java in the Cloud Age



Building

# Modular Cloud Apps with OSGi

O'REILLY®

Paul Bakker & Bert Ertman

A screenshot of the Amdatu website displayed in a browser window. The browser's address bar shows 'amdatu.org'. The website has a dark header with the 'amdatu' logo. A left-hand navigation menu lists various sections: HOME, Overview, News, GETTING STARTED (How to Use, Introduction to Modularity, Setting up the IDE, Creating a web app, Cloud deployment, Release management, Downloads), COMPONENTS (RESTful web services, Multi tenancy, Search, MongoDB, Blob stores), and GETTING INVOLVED (Contributors, Source, Mailing lists, Wiki, Issues). The main content area features the Amdatu logo and two sections: 'OSGi cloud components' and 'The Amdatu Way'. The footer contains copyright information for 2012 Amdatu, licensed under the Apache License, Version 2.0.

amdatu.org

amdatu.org

iPhone Dev Center | iGoogle | NU.nl | Apple | Yahoo! | Google Maps | YouTube | Wikipedia | News | Popular | Mac Stuff | Other Bookmarks

**amdatu**

HOME

- Overview
- News

GETTING STARTED

- How to Use
- Introduction to Modularity
- Setting up the IDE
- Creating a web app
- Cloud deployment
- Release management
- Downloads

COMPONENTS

- RESTful web services
- Multi tenancy
- Search
- MongoDB
- Blob stores

GETTING INVOLVED

- Contributors
- Source
- Mailing lists
- Wiki
- Issues

**Amdatu**

**OSGi cloud components**

Amdatu is an open source community effort focussed on bringing OSGi to the cloud. It contains components to create RESTful, scalable and distributed web applications that use NoSQL data stores, transparent multi-tenancy and much more.

**The Amdatu Way**

Amdatu is designed to be modular; you only use the components that you actually need for your application. You can use Amdatu components in any OSGi application, no matter how you build it. Following the Amdatu Way however you get a streamlined development and production flow which has been proven to work well. If you are new to OSGi you should have a look at the getting started guides provided on this website that will show you the Amdatu Way. If you have an existing application and are just looking for useful components, take a look at the components available.

Copyright © 2012 Amdatu. Licensed under the Apache License, Version 2.0.



## COURSE: Modular Application Development using OSGi and Amdatu

A Practical Guide to Modularity in the Cloud Age.

Java is an awesome platform and probably the best choice for any kind of enterprise (web) application today. But how do you design a system that can evolve for many years in production? And how do we run such systems in the cloud?

www.amdatu.com

Designing a system that can evolve over its lifetime is a hard problem. It is a hard problem because a modular design is often the only way to do it correctly. A modular architecture will help a lot to solve this problem. Modular design is a design of systems which enable you to update parts of a system without having to rebuild the whole system. The only modular approach for Java is OSGi. OSGi is an awesome technology to solve modularity and service-oriented architecture. Without these APIs you will have a hard time to actually build a platform. Unfortunately OSGi and Java enterprise technologies did not merge completely in the past. It was very frustrating to me both. But what if we were revolutionary in our architecture but also the creator of limited high-level (well-abstracted) APIs? The next question is how to deploy such applications in the cloud. Of course you could use one of the existing tools provided, but then basically deploy your old code to the cloud. This doesn't necessarily work well. How can you update a running application without creating the risk of it?

The up-to-date modular course enables you to move from a modular programming model to a modular design with some complex issues. It will present a practical and proven approach to building modular enterprise applications that can be deployed into a public/private cloud. The course will introduce OSGi, the Amdatu Open Source Cloud OSGi Components, and Apache ACE, an Open Source Provisioning Platform. We will not only focus on technical features, but also on architecture aspects like functional requirements, dependency management, design considerations, and testability.



## Course Contents

Web Services  
API Design Considerations  
OSGi  
OSGi Clients  
OSGi Containers  
OSGi

OSGi Strategies

OSGi

OSGi

OSGi

OSGi (Microservices)

OSGi

OSGi

OSGi

OSGi

OSGi

OSGi

OSGi

OSGi

OSGi

OSGi

OSGi

OSGi

OSGi

OSGi

OSGi

OSGi

OSGi

OSGi

OSGi

OSGi

OSGi

OSGi

OSGi





cloud provisioning

<http://ace.apache.org/>



cloud OSGi services

<http://www.amdatu.org/>



**Amdatu**

Bert Ertman

[bert.ertman@luminis.eu](mailto:bert.ertman@luminis.eu)



@BertErtman

Eclipse OSGi plugin

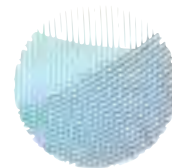
<http://bndtools.org/>



That's us

<http://luminis.eu/>

**luminis**



Paul Bakker

[paul.bakker@luminis.eu](mailto:paul.bakker@luminis.eu)



@pbakker



Dank u

Merci

Danke

Mahalo

Obrigado

Grazie

Gracias

Thank  
you

Tak