

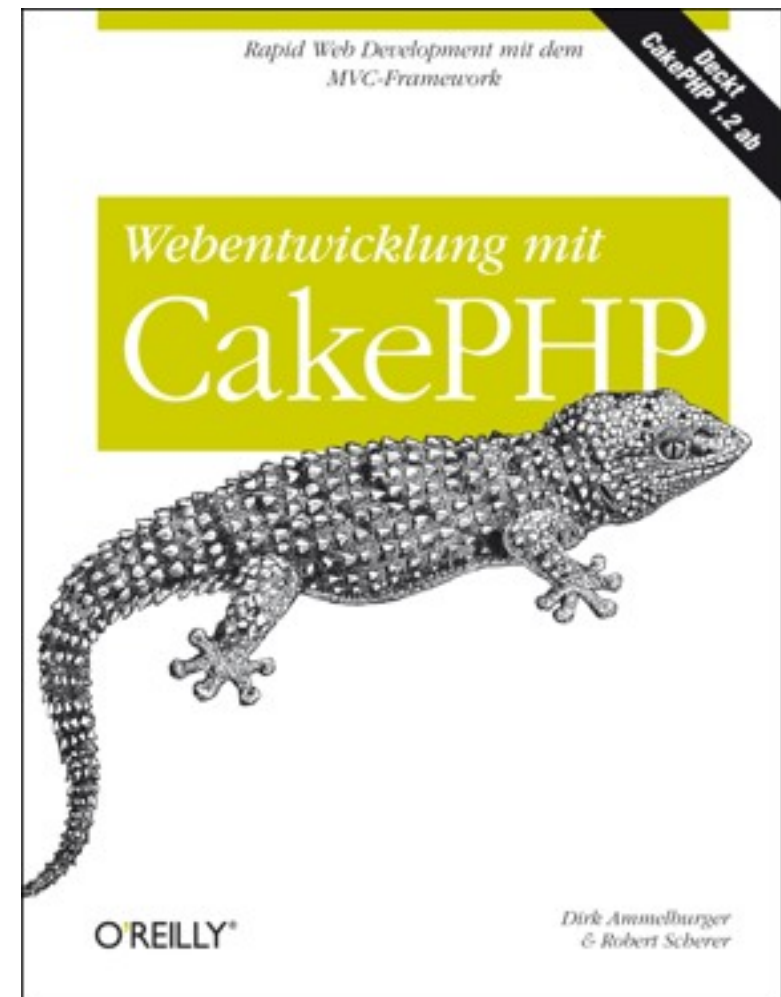
Multi-tenancy with CakePHP

Robert Scherer
M-Invent GmbH

Komplexe
Anforderungen.
Individuelle
Lösungen.

About me

- developer and consultant
- head of the CakePHP business unit of M-Invent GmbH
- author of the first german book about CakePHP, published on O'Reilly



About M-Invent GmbH

- Munich based IT and management consultancy
- Mainly active in the insurance and banking industry
- Using CakePHP in addition to Java Technology since mid-2008

What we'll cover

- What is multi-tenancy?
- What are the challenges to solve?
- How we solved it in a project with CakePHP

What is multi-tenancy?

"**Multitenancy** refers to a principle in software architecture where a single instance of the software runs on a software-as-a-service (SaaS) vendor's servers, serving multiple client organizations (tenants)."

(Wikipedia)

In practice:

- “hardcore whitelabeling”
- multiple clients on one installation of the application
- each client needs customization in the software
- goal is a single code base and a clean implementation of the tenants customizations

Main challenges in customization

- Differences in the functionality
- Workflow / modification of functionality
- Separation of data
- Branding

The requirement

- multi-tenant application for a major insurance company
- purpose is distributing insurance policies
- currently about 30 different mandators
- some mandators with major customization needs, some pretty standard
- configuring mandators without using IT resources

The legacy

Legacy application:

- Spaghetti PHP at its finest
- about 40% obsolete tables in 3 databases
- 8 different code bases
- multi-tenancy was solved with the copy & paste pattern

The system

Three entities in the system:

- Insurance vendor (project owner)
 - Mandators (e.g. auto companies)
 - Dealers (e.g. car dealer)

- each entity has its user groups (administrators, employees, technical user)

- solved with Cake Auth and ACL

The approach

- make the system as modular as possible
- make the system as configurable as possible
- implement the customizations as generic as possible

Modules and Addons

Module

Consists of at least:

- A module class
- A module controller

contains

Addon

Consists of:

- an addon class
- elements

- modules are groups of functionality
- addons are extensions to the functionality of modules
- modules can be assigned to mandators
- addons can be assigned, too

Keep it generic

What you don't want:

```
if($_SESSION['company'] == 'Foo Inc.') {  
    // do stuff  
}
```

What you want:

```
if(Mandator::hasAddon('generate_insurance_number')) {  
    $module->addon('generate_insurance_number')->doStuff();  
}
```

Form customization

Fahrzeugdaten

Fahrzeugtyp

Hersteller

Modell

Erstzulassung

Fahrzeugdaten

Hersteller
 ⓘ

Modell
 ⓘ

Erstzulassung

- same form, different functionality
- addons can act in all MVC layers

Configure mandators

Mobilitätsgarantie

Modul für Mobilitätsgarantien

aktivieren

Konfiguration einblenden

Addons

Herstellerauswahl

AddOn zu Mobi zur Auswahl von Hersteller/Modell

Fahrzeugtyp-Auswahl

AddOn zu Mobi zur Auswahl von Fahrzeugtyp

Konfiguration einblenden

Wartungsintervall

Wartungsintervall in KM

Konfiguration einblenden

Nächste Wartung

Auswahl nächste Wartung in X Monaten oder Y km

Konfiguration einblenden

Grund d. Aktivierung

Grund für die Aktivierung der Mobilitätsgarantie

Konfiguration einblenden

Mobi Edit Lock

Verhindert das Editieren von Mobilitätsgarantien nach einer bestimmten Zeit.

Konfiguration einblenden

Mobi Delete Lock

Verhindert das Löschen von Mobilitätsgarantien nach einer bestimmten Zeit.

Konfiguration einblenden

Configure mandators

Mobilitätsgarantie

Modul für Mobilitätsgarantien

aktivieren

Konfiguration einblenden

Addons

Herstellerauswahl

AddOn zu Mobi zur Auswahl von Hersteller/Modell

Fahrzeugtyp-Auswahl

AddOn zu Mobi zur Auswahl von Fahrzeugtyp

Konfiguration ausblenden

Konfiguration

Fahrzeugtypen

auf Default zurücksetzen

Mögliche Optionen für Fahrzeugtypen. Format "key1:Text1,key2:Text2..."

Wartungsintervall

Wartungsintervall in KM

Konfiguration ausblenden

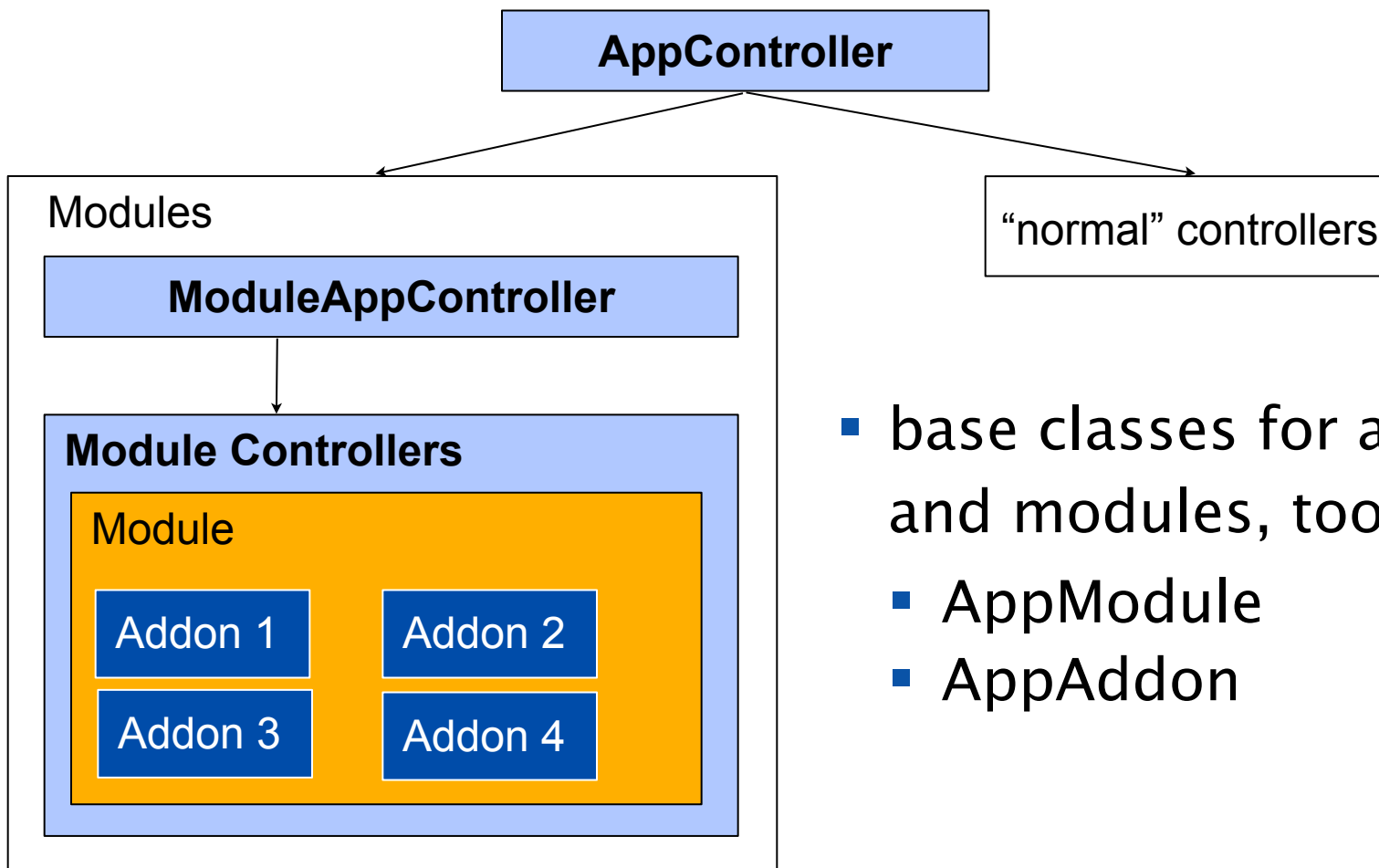
Konfiguration

Wartungsintervalle

auf Default zurücksetzen

Mögliche Optionen für Wartungsintervalle

Structure



- base classes for addons and modules, too:
 - AppModule
 - AppAddon

Configuration

- Very powerful configuration system
- System-wide configuration or feature-specific configuration (modules/addons)
- every configuration has its default value, degraded from the upper level

Priority of configurations

1. System default
2. Mandator's configuration
3. Dealer's configuration

Additional configuration

Konfiguration

Wording

Mobilitätsgarantie Artikel

 auf Default zurücksetzen

Definition des Wording von Mobilitätsgarantie, Artikel, z.B. die

Mobilitätsgarantie plural

 auf Default zurücksetzen

Definition des Wording von Mobilitätsgarantie, plural, z.B. Mobilitäten

Mobilitätsgarantie singular

 auf Default zurücksetzen

Definition des Wording von Mobilitätsgarantie, singular, z.B. Mobilität

Data separation

- we didn't use ACL for row-level permissions
- parts of it happening at controller level, other parts in the models
- reason are very different requirements and circumstances across the system

Branding

- loading mandator-specific CSS
- CakePHP themes

Shameless plug

- Are you a motivated CakePHP developer, based in Germany and looking for a project? Get to us!
- Need professional help developing CakePHP applications?
We can help!

robert.scherer@m-invent.de

End

```
if(!empty($questions)) {  
    $this->goAhead();  
} else {  
    die('Thanks!');  
}
```