



## One Vision – One Platform

Eclipse als Plattform für die GUI Integration



*Besser begleitet.*

**PostFinance** 



# Agenda

- PostFinance
- Eclipse Technologie
- Erkenntnisse/Erfahrungen
- Q&A

*Remo Lötscher*

Technologiemanager GUI-FW

Project Lead „Yet Another RCP Inspector“ und „RCPForms“

Seit 2007 im PFGUI-Team

[remo.loetscher@postfinance.ch](mailto:remo.loetscher@postfinance.ch)

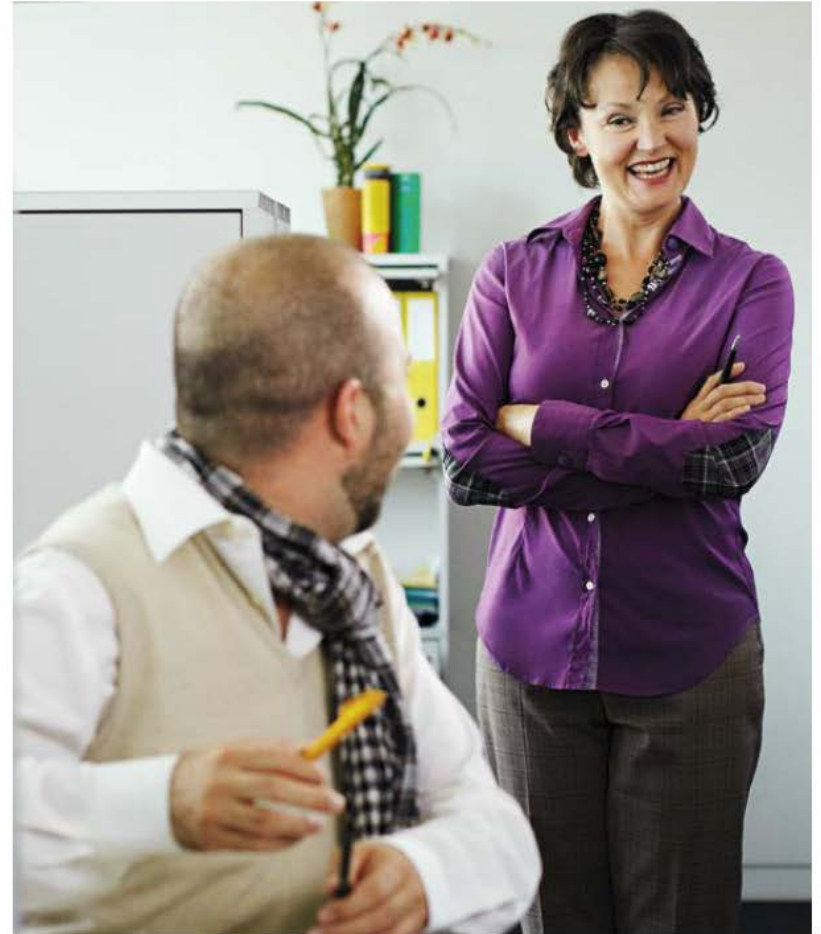
# PostFinance stellt sich vor...



# Als erfolgreiche Finanzdienstleisterin prägen wir den Schweizer Retailmarkt

## Wer wir sind

- Eines der führenden **Retail-finanzinstitute** der Schweiz
- Nummer eins im Schweizer **Zahlungsverkehr**
- Die ideale Partnerin für Kundinnen und Kunden, die ihre Finanzen **selbstständig verwalten**



# Unseren Privat- und Geschäftskunden offerieren wir umfassende Finanzdienstleistungen.

## Was wir bieten

- Einfache und attraktive Finanzprodukte in den Bereichen **Zahlen, Sparen, Vorsorgen, Anlegen und Finanzieren**
- Bester Service **rund um die Uhr** online und telefonisch
- Privatkunden beraten **wir in den Filialen**
- Geschäftskunden beraten **wir vor Ort im Betrieb**





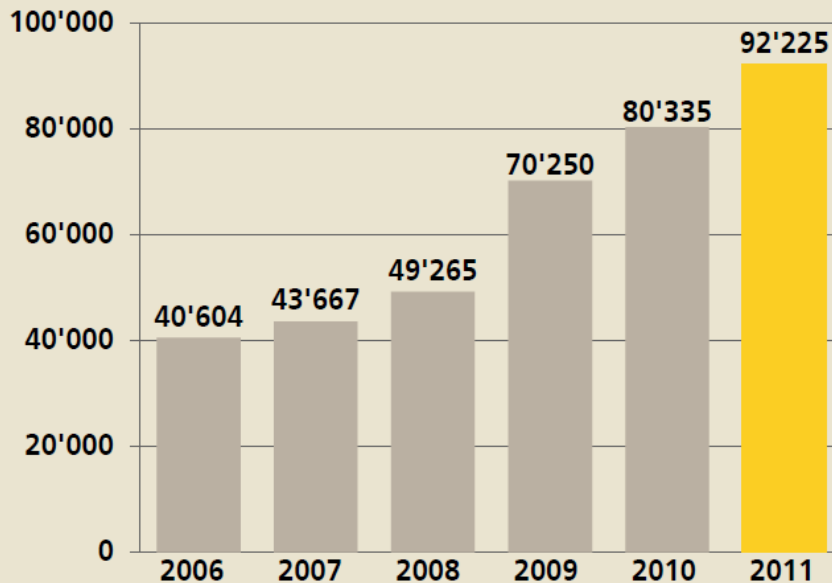
# Wir wollen weiter wachsen und uns als eines der führenden Schweizer Finanzinstitute etablieren.

## Kundenzuwachs

Wir betreuen **2,8 Mio. Kundinnen und Kunden**. Das sind 300 000 mehr als im Jahr 2008. Wir führen **4,4 Mio. Konten**. Das sind 800 000 mehr als im Jahr 2008.

Stand 1. Quartal 2012

## Kundenvermögen in Mio. CHF



Stand: 31.12.2011

## Mitarbeiterzuwachs

Bei uns arbeiten knapp **4000 Mitarbeiterinnen und Mitarbeiter**. Das sind 600 mehr als im Jahr 2008.

Stand 1. Quartal 2012

## Marktführerin

Wir sind die **Nummer 1** im Schweizer Zahlungsverkehr und eines der führenden Retailfinanzinstitute.

## E-Finance

**Über 1,4 Mio. Kundinnen und Kunden** erledigen alles rund ums Geld bereits online.

Stand 1. Quartal 2012

# Eclipse RCP als Laufzeitumgebung bei PostFinance

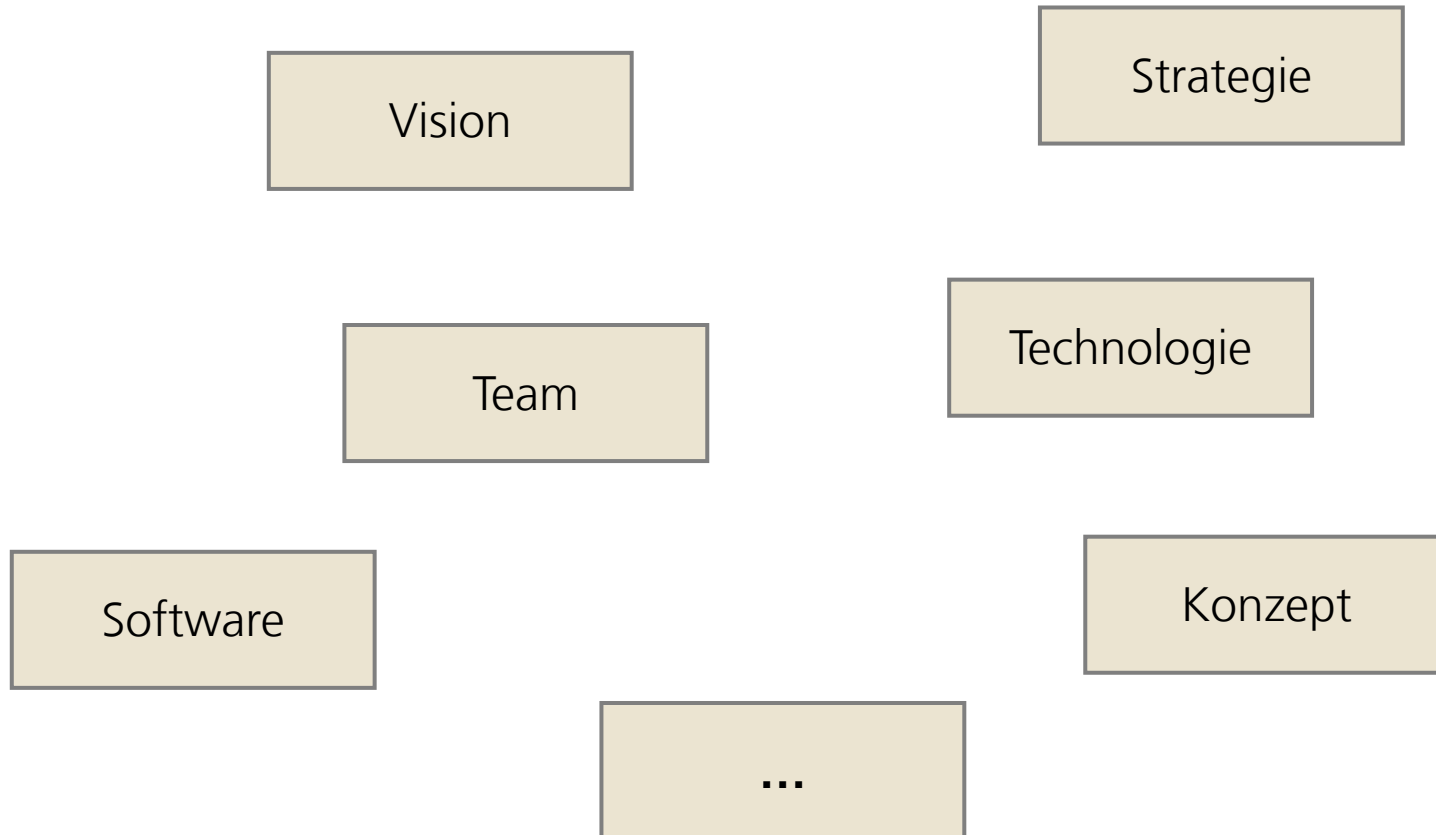


**P**ost  
**F**inance  
**G**raphical  
**U**ser  
**I**nterface





# PFGUI



„Pro Mitarbeiterin/Mitarbeiter gibt es eine GUI-Applikation um die tägliche Arbeit zu erledigen“



# PFGUI

## Integration auf GUI Layer

Stufe I:	Stufe II:	Stufe III:
GUI-Integration durch Benutzer	GUI-Integration durch Kontext-Koordinator	Volle GUI-Integration
<b>Standardprodukte</b> <ul style="list-style-type: none"> <li>• verschiedener Hersteller</li> <li>• <b>ohne</b> PF-spezifische Anpassungen</li> </ul>	<b>Standardprodukte</b> <ul style="list-style-type: none"> <li>• verschiedener Hersteller</li> <li>• <b>mit</b> PF-spezifischer Anpassung</li> </ul>	<b>PF-spezifisches GUI</b> <ul style="list-style-type: none"> <li>• <b>„aus einem Guss“</b></li> <li>• optimiert für Unterstützung der <b>Geschäftsprozesse</b></li> </ul>

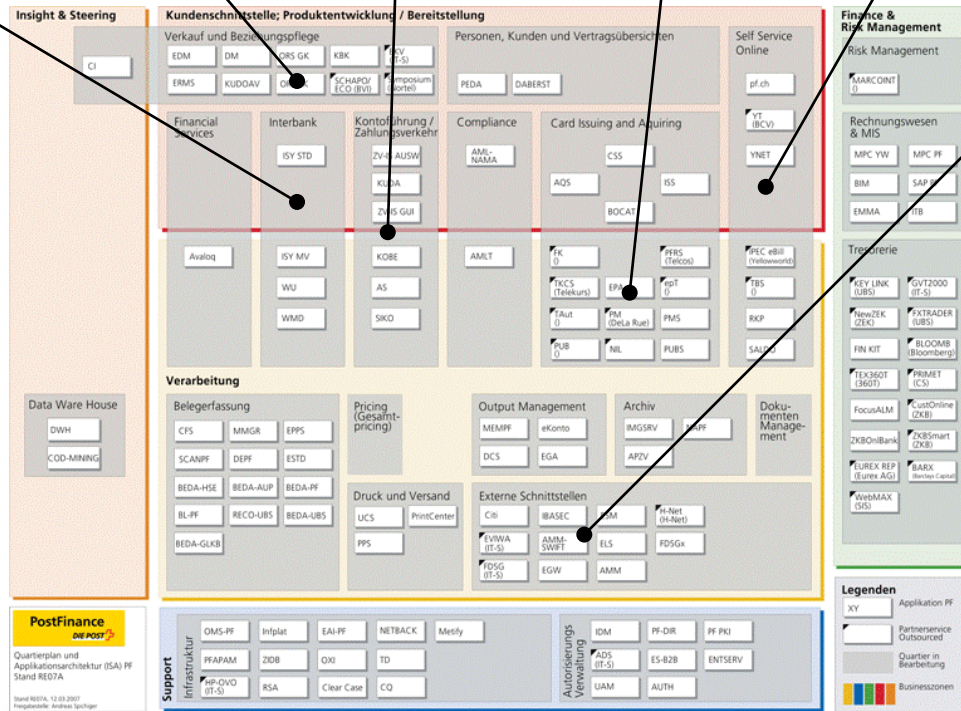
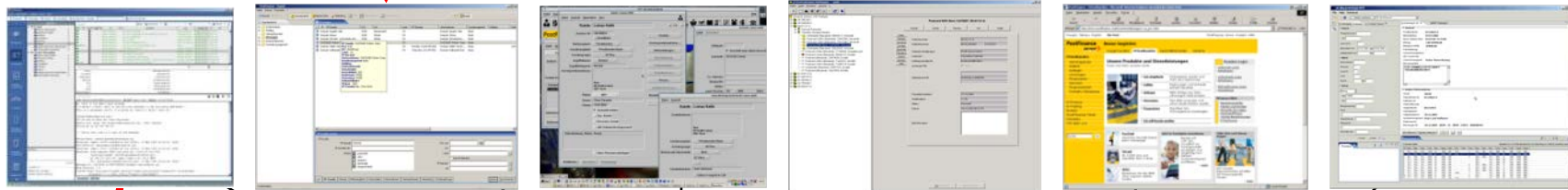
2003

2008

2010

# PFGUI

## Integration auf GUI Layer – Stufe I (Ist-Zustand 2003)





# PFGUI

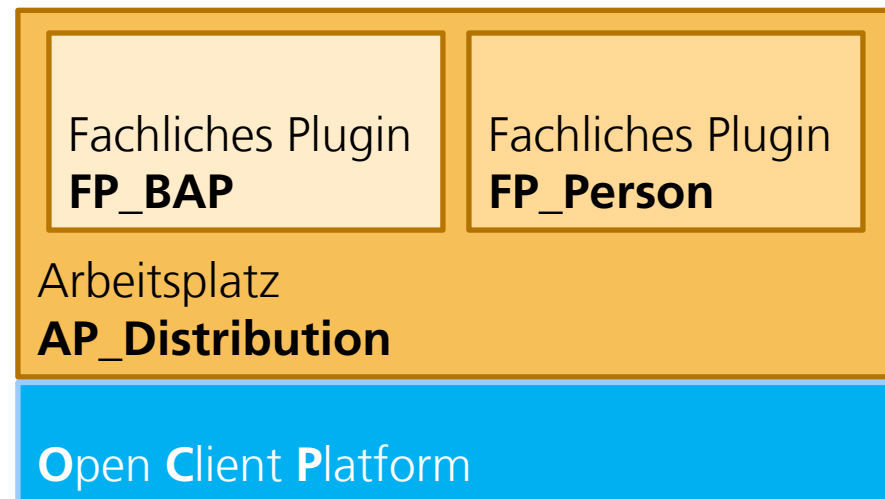
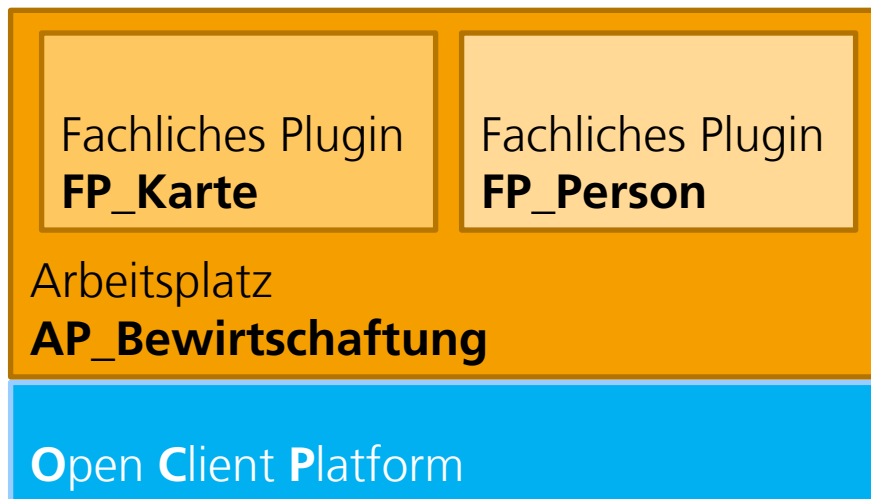
## Begrifflichkeiten

### Fachliches Plugin

Ein Fachliches Plugin ist eine *in sich abgeschlossene GUI-Komponente* für die Bearbeitung von Daten. Dieses umfasst z.B. eine Suche, das Anzeigen und Bearbeiten (Mutieren, Löschen, Neuerfassen) von Daten. Somit gehören zu Fachlichen Plugins Editoren, Views (Anzeigefenster) und Dialoge. Ein Fachliches Plugin kann in mehreren Arbeitsplätzen wiederverwendet werden.

### Arbeitsplatz

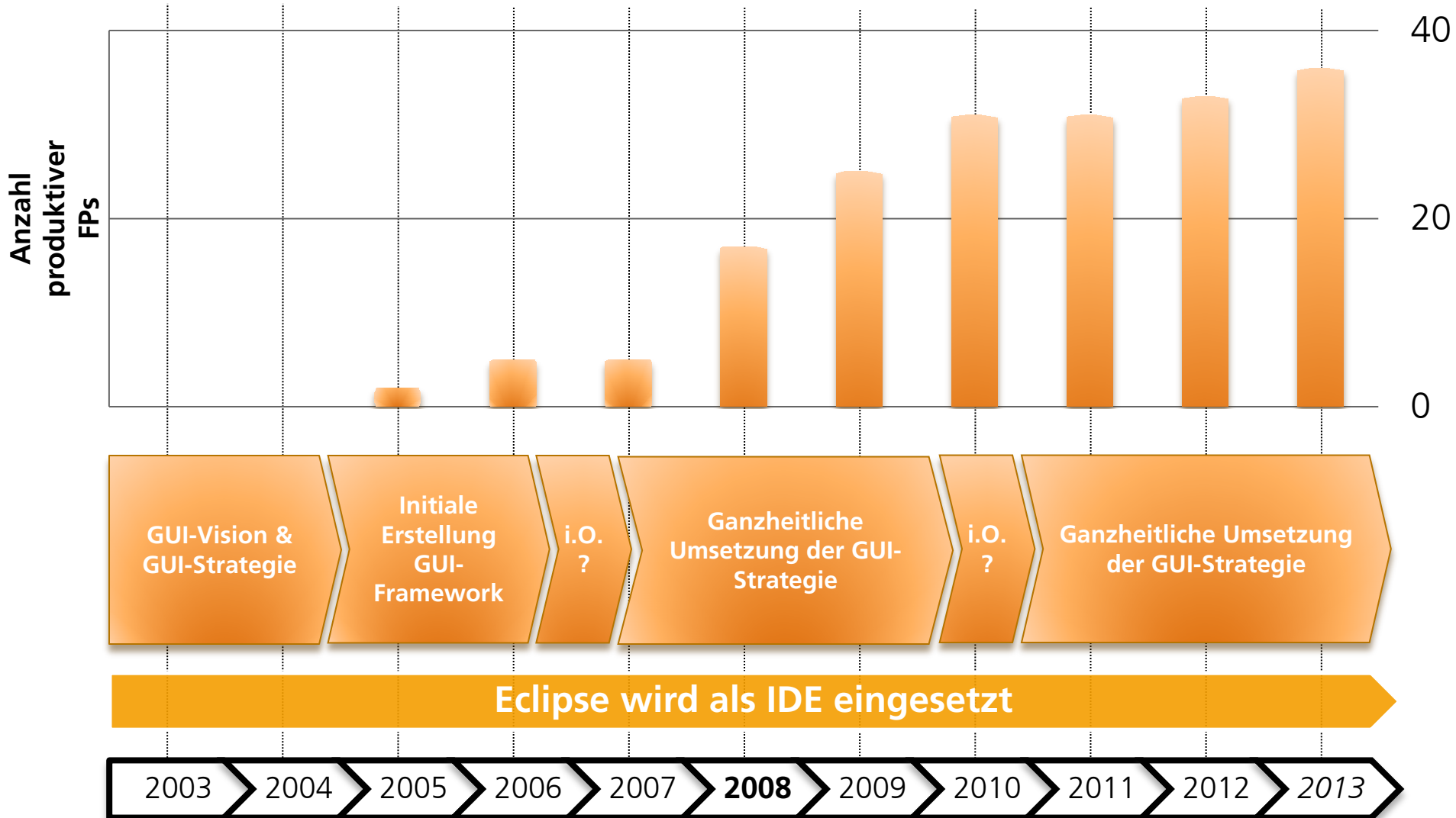
Ein Arbeitsplatz ist eine *Zusammenstellung aus Fachlichen Plugins* (Container für Fachliche Plugins, RCP Applikation).





# PFGUI

## Einsatz von GUI-Applikationen auf Basis Eclipse RCP (Stand 10.2012)





# PFGUI

## Eclipse RCP Clients

### Businesslogik: Arbeitsplätze, Fachliche Plugins

#### PFGUI (Team, Framework, Strategie, ...)

**OCP Search** Komponente

**RCPForms** – Widget-Factory

**OCP Support** Komponente

**OCP Security** Komponente

**OCP Autorisierung** Komponente

**OCP Remote** Komponente

**Diverse Erweiterungen**

**Public Service Interfaces**

Open Client Platform

Eclipse (RCP)



**Vorgaben** Techn. und UI-Guidelines

**Tools** Iconverwaltung, Help, NL-Utility

**Know-How-Pflege**

Intranetseite, Codecamps, Coaching, Samples

**Organisation/Koordination**

Usergroup, Fachtechnische Organisation, Betrieb

Ein **FP** ist das Frontend (GUI) zu einer PF-Applikation:

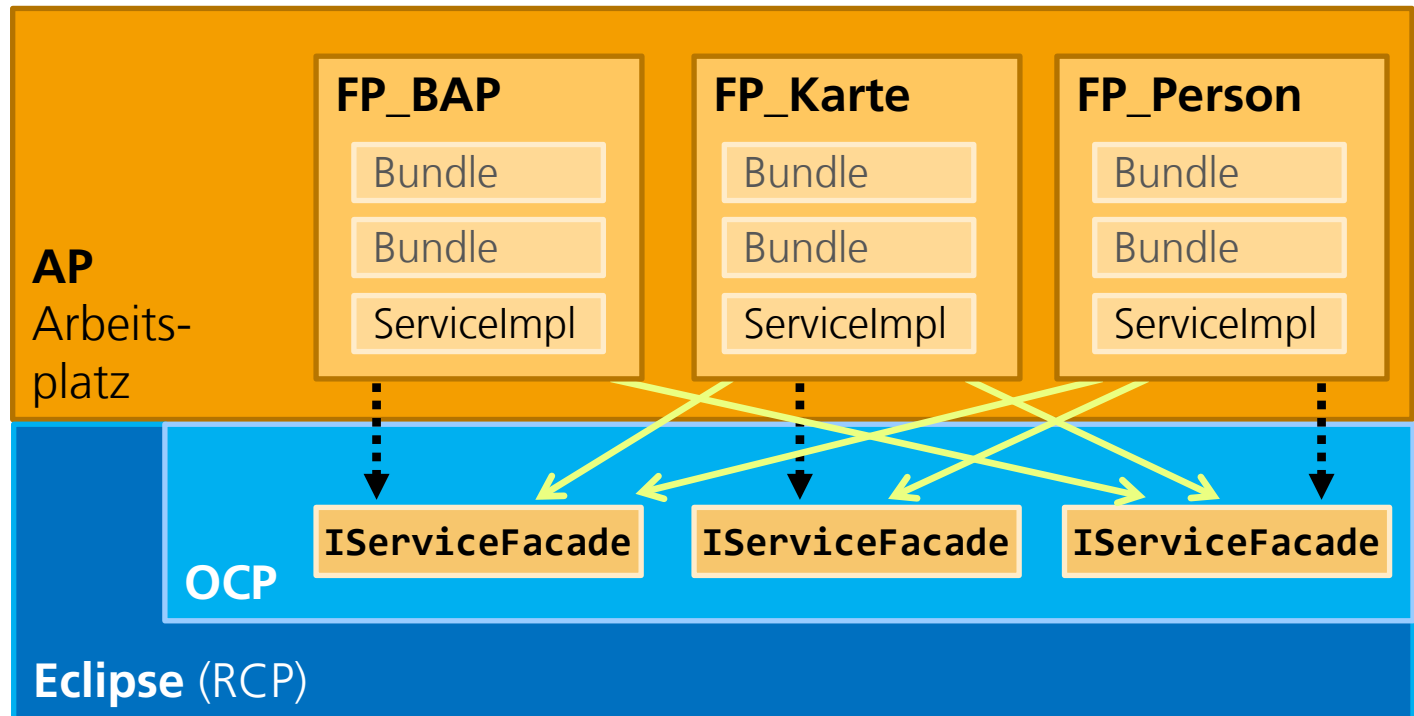
– FP bieten OSGi Services an

– Ein FP kann Funktionen aus anderen FPs aufrufen →

– Person

– Karte

– Beratung



# Beispiel – AP Distribution (RE12C)

Maturitätsstufe III live

**Komponenten**  
aus FP\_XYZ...0

**Komponenten**  
aus FP\_Person

**Komponenten**  
aus FP\_Karte

The screenshot displays the SAP GUI interface for a search operation. The main window is titled 'TEST AP Distribution' and shows search results for a person and a PostFinance Card. The interface is annotated with colored boxes and lines pointing to specific components:

- Red box:** Encloses the search criteria and results table.
- Blue box:** Encloses the search criteria section, including 'Suche' and 'Suchen nach...'. A line points from this box to the 'Komponenten aus FP\_XYZ...0' label.
- Yellow box:** Encloses the search results table, including the 'Personensuche: 1 Resultat(e)' header and the table data. A line points from this box to the 'Komponenten aus FP\_Person' label.
- Green box:** Encloses the details view for the 'PostFinance Card', including the 'PostFinance Card' header and the 'Produkteigenschaften' section. A line points from this box to the 'Komponenten aus FP\_Karte' label.

The search results table shows the following data:

Partner-Nr.	Name	Vorname	Geburtsdatum	Nationalität	PLZ	Ort	Land	Person ist	Strasse	Hausnummer	Personal-Nr.	hat Ba
1003...	Bernasconi	Maria		Schweiz	...	...	Schweiz	Kunde	...	1		Nein

**Search Komponente**  
Umgesetzt in PFGUI OCP  
(Extension Point)

# Erkenntnisse



# Lessons Learned

## Organisatorisch

### Highlights:



- Know-How-Management/Experten PFGUI CC + FW Team
- Releaseplan / Releasetrain
- Vorgaben nur so viel wie sinnvoll (Dokumente)
- Organisationskonzept
- Fach-Technische Koordination
- Austausch (UserGroup, Collaboration Platform, Ressourcenpooling)

### Lowlights:



- Qualitätssicherung QoS
- Verwendung Eclipse APIs/Konzepte
- Funktionalität (Wiederverwendung)
- Performance/Memory
- Vorgaben nicht eingehalten
- Usability Engineer: spät eingesetzt
- Ressourcen
- Testing : nur FP, nicht aber innerhalb eines AP

# Lessons Learned

## Technisch

### Highlights:



- Entkopplung durch Schnittstellen (Public Plugin) und IoC
- Kopplung FW-Team  $\leftrightarrow$  Business Logik:
  - “Eat your own dog food” vs “Elfenbeinturm”
- Eclipse & OSGi Plattform
- Gemeinsame Laufzeitumgebung: Singletons, shared resources, resource handles
  - anspruchsvolle Entwicklung

### Lowlights:



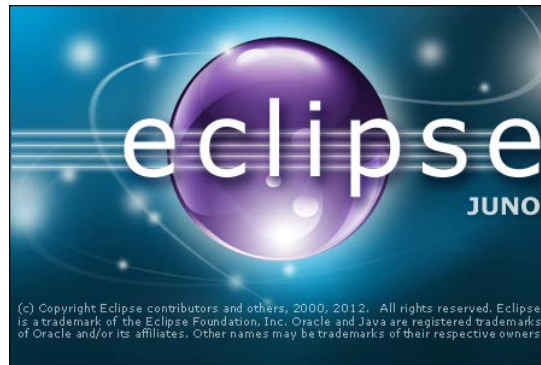
- Keine konsequente Wiederverwendung (Widget-Toolkits, Forms):
  - MVC → x Modelle: POJO, EMF, Beans, DTOs, xText-Generat ...
  - erhöhte Weiterentwicklungs- und Betriebskosten
- Keine Eclipse Autoupdate Funktionalität
- Keine einheitliche Deployment- und Buildinfrastruktur
- WLS → Buddy-Policy  
“*Buddies are evil*”

# Ausblick





- Konsolidierung
- Weiterentwicklung
- Zentrale Buildinfrastruktur
- Integration von externen UIs
  - Web
  - Swing
  - ...
- e4/Juno



# Fragen?



Besten Dank für Ihre Aufmerksamkeit!

