

# Architektur der OpenMDM 5 Anwendung

Stand: 22. September 2015

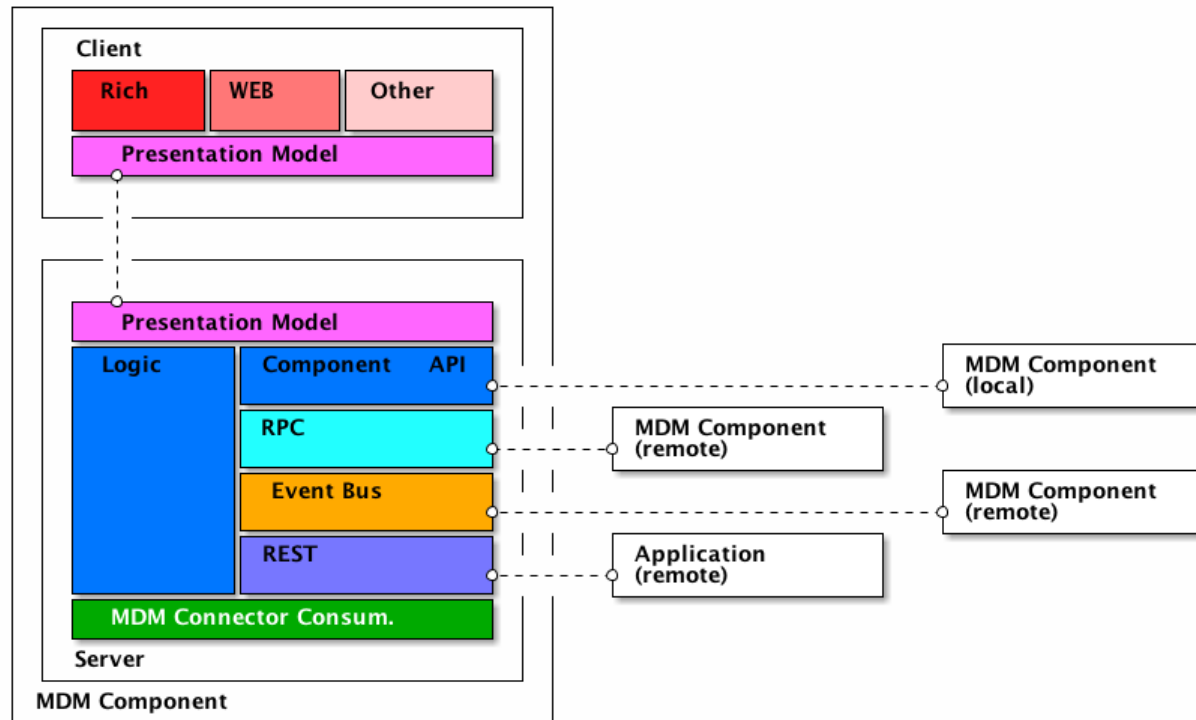
Franz-Josef Basler

Sibylle Peter

[delivering end-user happiness]

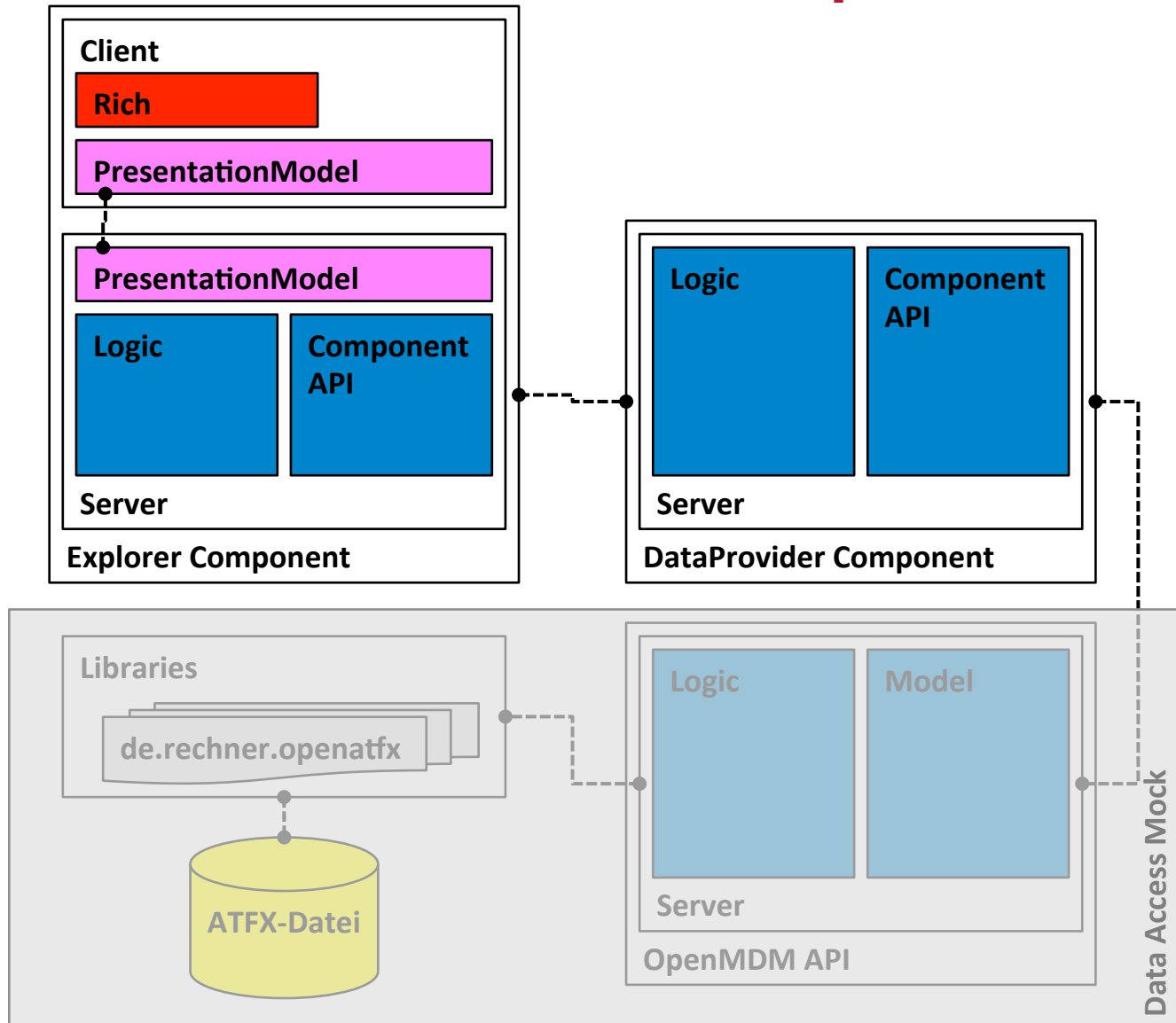
**canoo**

# Vorgesehene Architektur



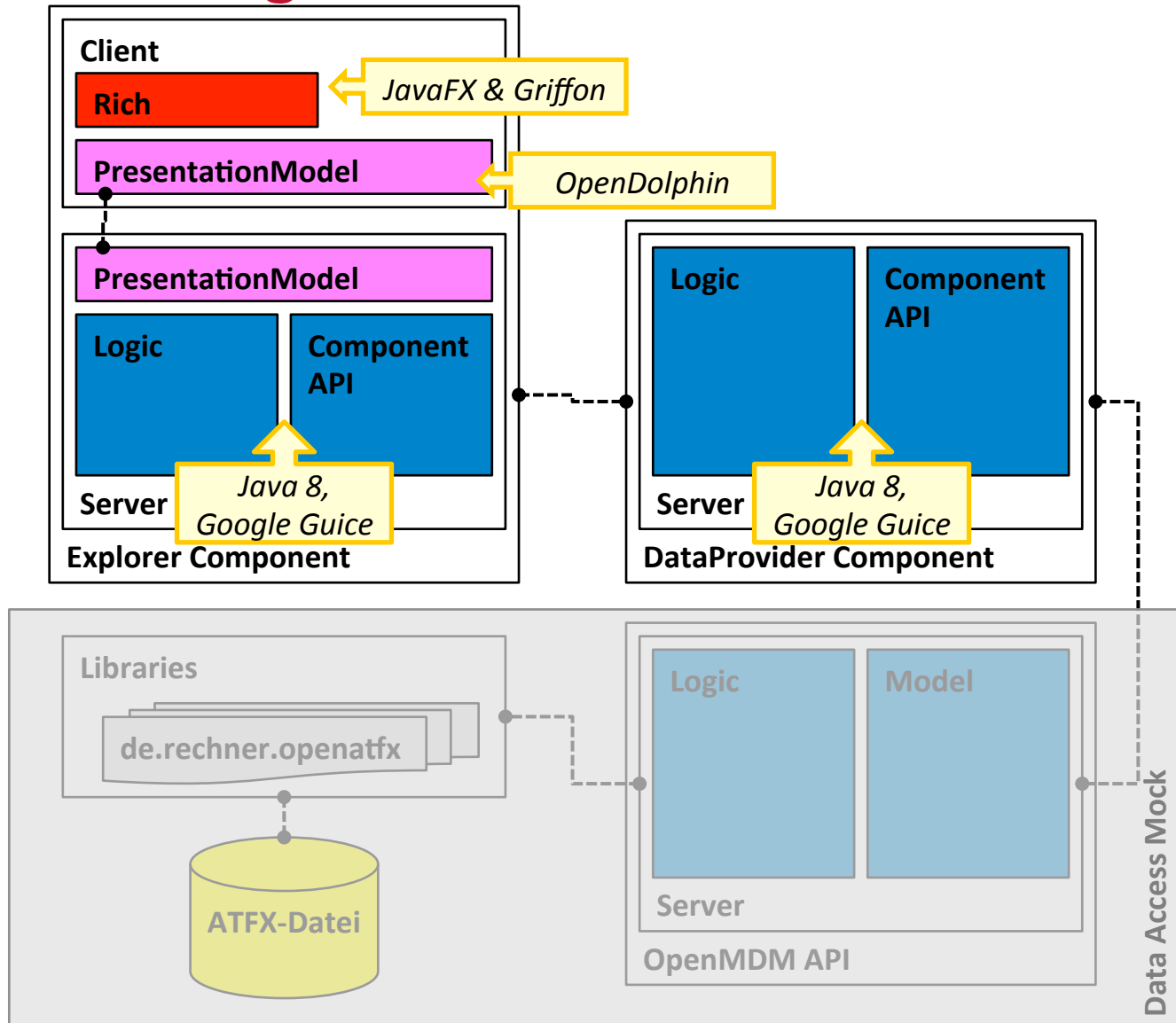
Je Komponente gemäß **OpenMDM® 5 Architektur Spezifikation**. Bei headless Komponenten entfällt der Client Teil.

# Aktuelle Architektur des Explorers



[delivering end-user happiness]

# Technologie-Einsatz

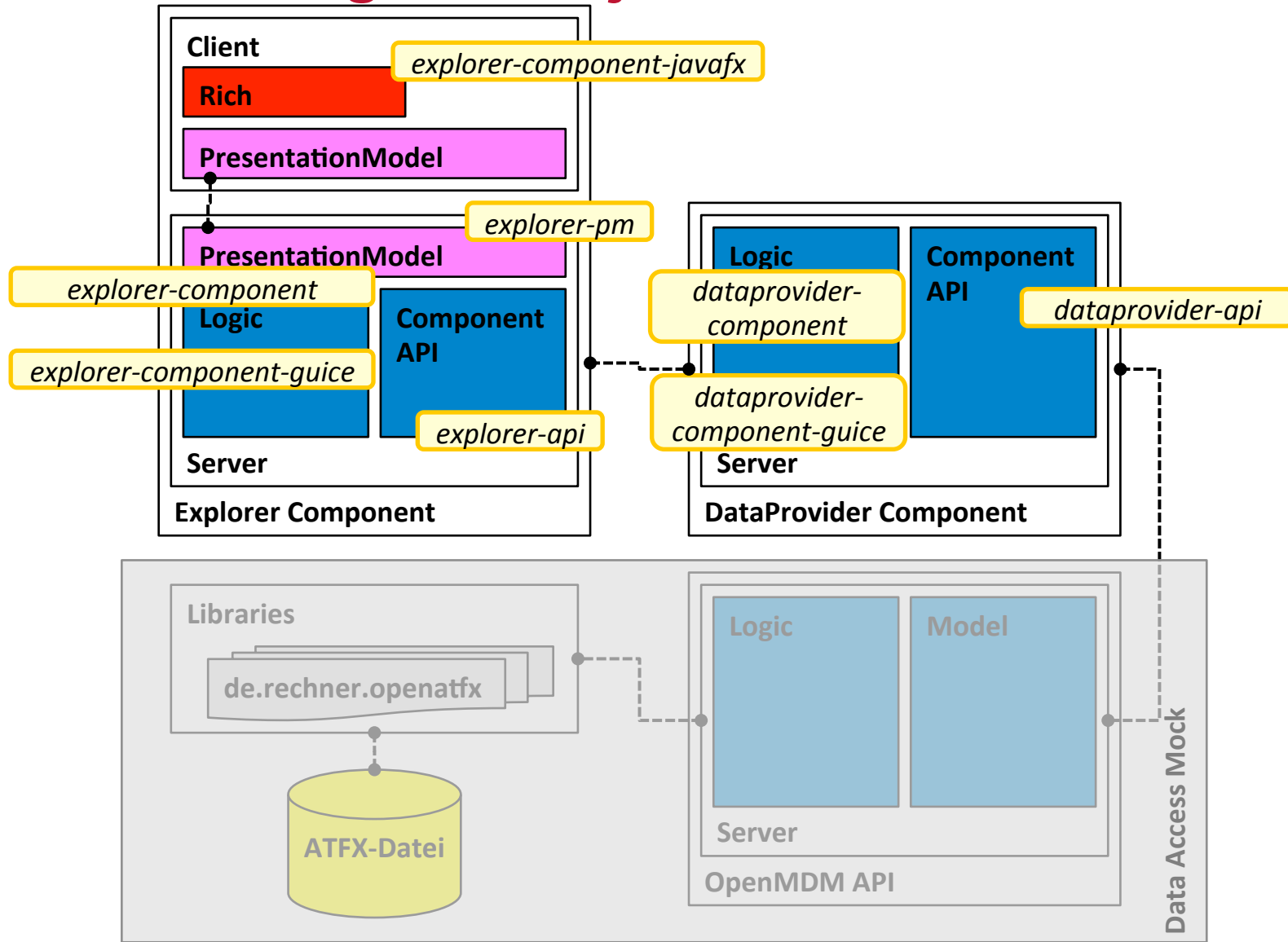


[ delivering end-user happiness ]

# Technologie-Einsatz

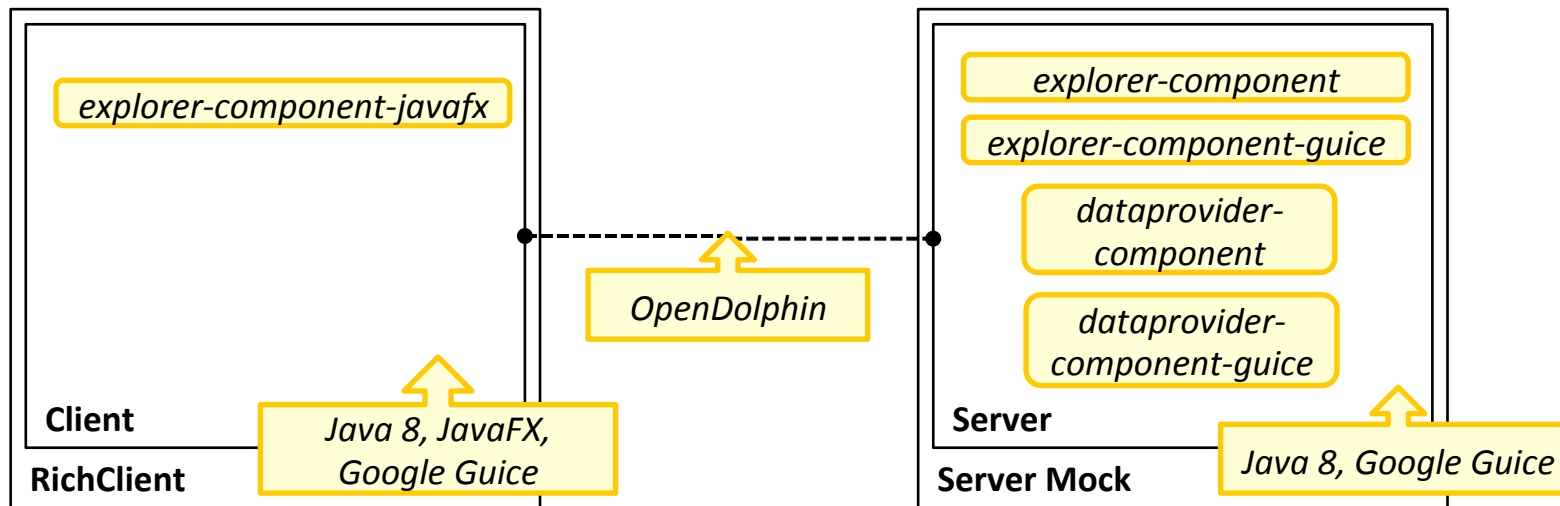
- Java 8 (Allgemeine Programmiersprache)
- Google Guice (Dependency Injection)
- Griffon (Client-MVC-Framework)
- JavaFX (Graphische Oberfläche)
- OpenDolphin (Client-Server-Kommunikation)
- Gradle (Build-Automatisierung)
- JaCoCo (Testabdeckung)
- TestFX (JavaFX Integrationstests)
- Spock (Unit-Tests in Groovy)

# Verteilung der Projektmodule



[ delivering end-user happiness ]

# Deployment



Jetty wird für den Server Mock verwendet.

Der RichClient stellt ein Applikationsfenster bereit, in das alle anderen Inhalte eingefügt werden können. Hierbei kommt ebenfalls das Griffon Framework zum Einsatz (MVC). Die Client-Server-Kommunikation erfolgt durch OpenDolphin Presentationmodels, welche via asynchronen HTTP POST übertragen werden.

Gestartet werden Client und Server jeweils mit \*.bat Skripten.

Nähere Informationen finden sich in den README.adoc Dateien im Root-Verzeichnis der Projekte.

# Nutzung Eclipse Infrastruktur

- Hudson (Automatisierte Builds)
- Sonar (Statische Quellcode Analyse)
- Release Repository (Artefakt Ablage)
- Snapshot Repository (Artefakt Ablage)