

CAS Software AG
Erfolg durch Customer Excellence



Entwicklung einer SaaS-Lösung mit Standardtechnologien – ein Praxisbericht



VKSI-Sneak-Preview „Entwickeln für die Cloud“
Markus Bauer

Agenda



- Einführung
 - CAS, SaaS und PIA
- Architektur
 - Technische Architektur, Deployment
 - Logische Architektur
 - Technologien
- Ausgewählte Schwerpunkte
 - Anpassbarkeit durch Metadaten
 - Integrationsfähigkeit, Offenheit
 - Sicherheit
 - Qualität, Monitoring
- Randnotizen
- Zusammenfassung

CAS Software AG

– CRM-Produkte für den Mittelstand



- Über 200.000 Anwender weltweit
- 145 nationale, 60 internationale Partner in 25 Ländern



- 39 Mio. Euro* Umsatz
- Rund 220 Mitarbeiter, ca. 400 CAS-Gruppe*



- Beteiligungen:



*Beteiligungsunternehmen anteilig berücksichtigt

CAS Produktportfolio



CRM Standard-Lösungen genesisWorld, PIA + Module



CRM Branchen- und Speziallösungen

CAS Drive
CAS netWorks
CAS IT Services
CAS Consulting
CAS Research
CAS Engineering

Lösungen für Hochschulen und Schulen

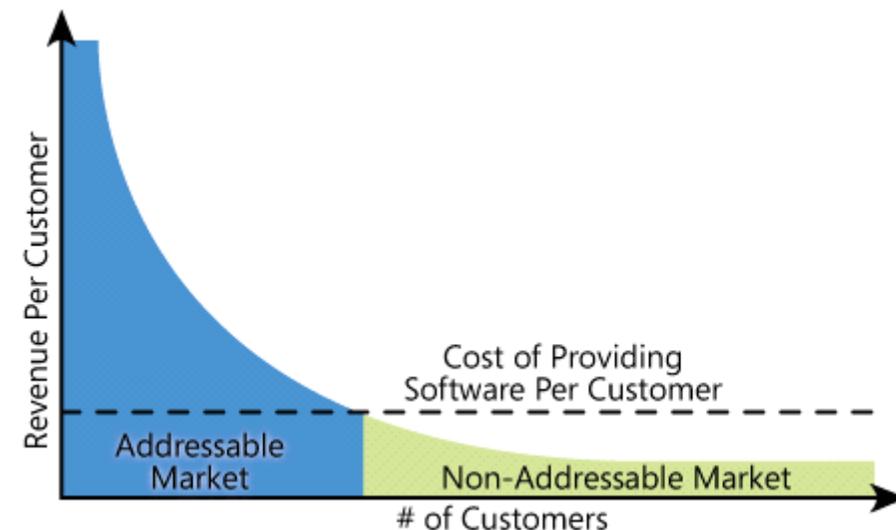
CAS Campus
CAS Alumni
CAS Platon
CAS Platon Intranet

+ Mehrwertdienste und CRM Individuallösungen

Warum SaaS?



- Grundidee: Ausnutzung von Skaleneffekten
- Aus Kundensicht:
 - Geringe TCO: geringe Anfangsinvestitionen (Hardware/Software), keine Aufwände für sicheren Betrieb, Backup, Upgrade
 - Vereinfachte Planung: Lösung wächst dynamisch nach Bedarf
- Aus Anbietersicht:
 - Größeres Marktpotenzial („Selling to the long tail“)
 - Stetige Einkünfte (Mietmodell)
 - Kostenreduktion durch einheitliche Releasestände

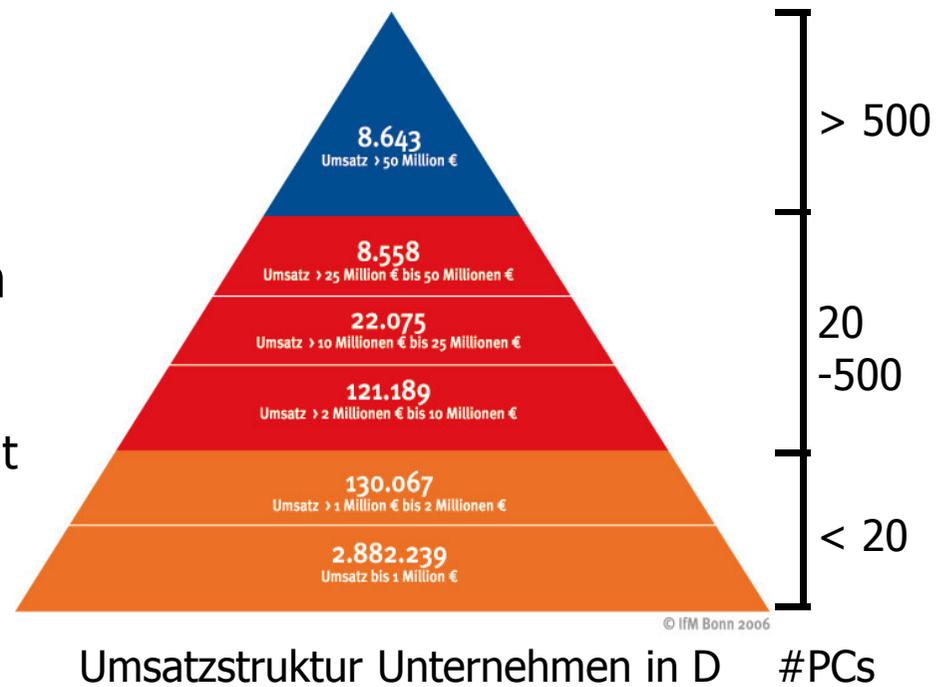


Quelle: Rainer Stropek, Microsoft Software Architecture Conference, Vienna 2010

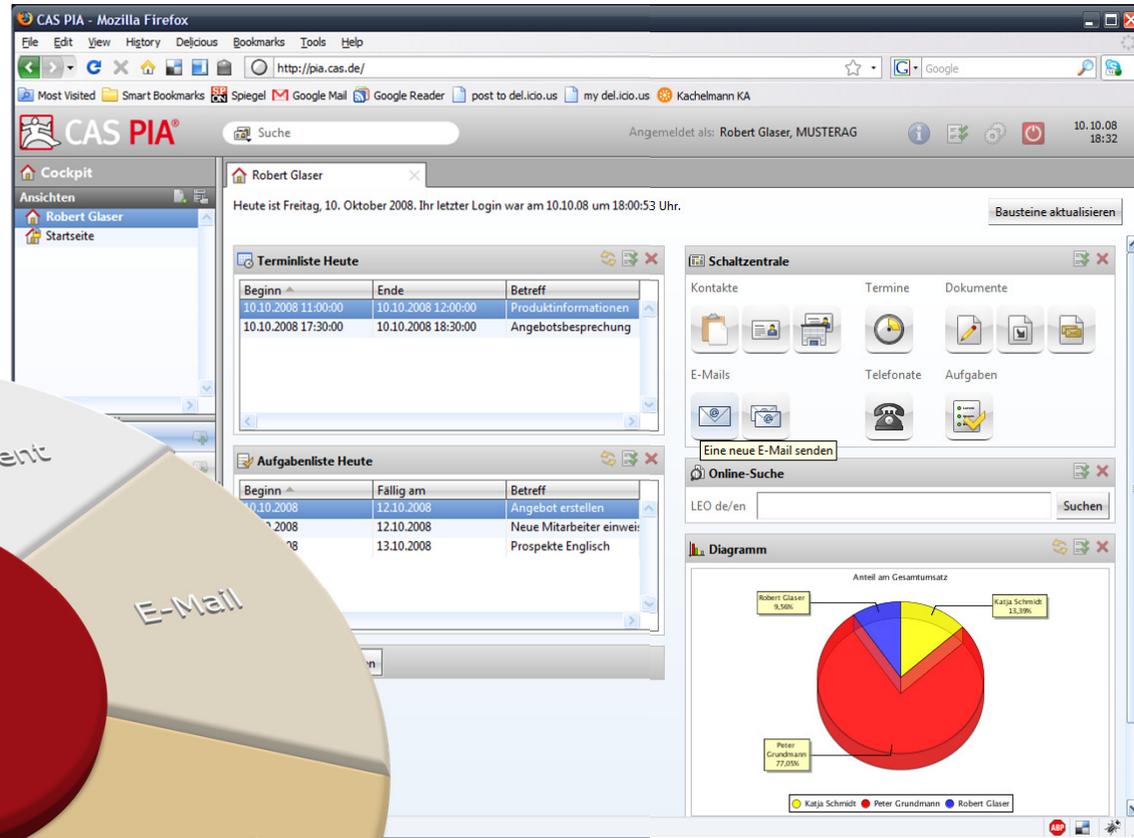
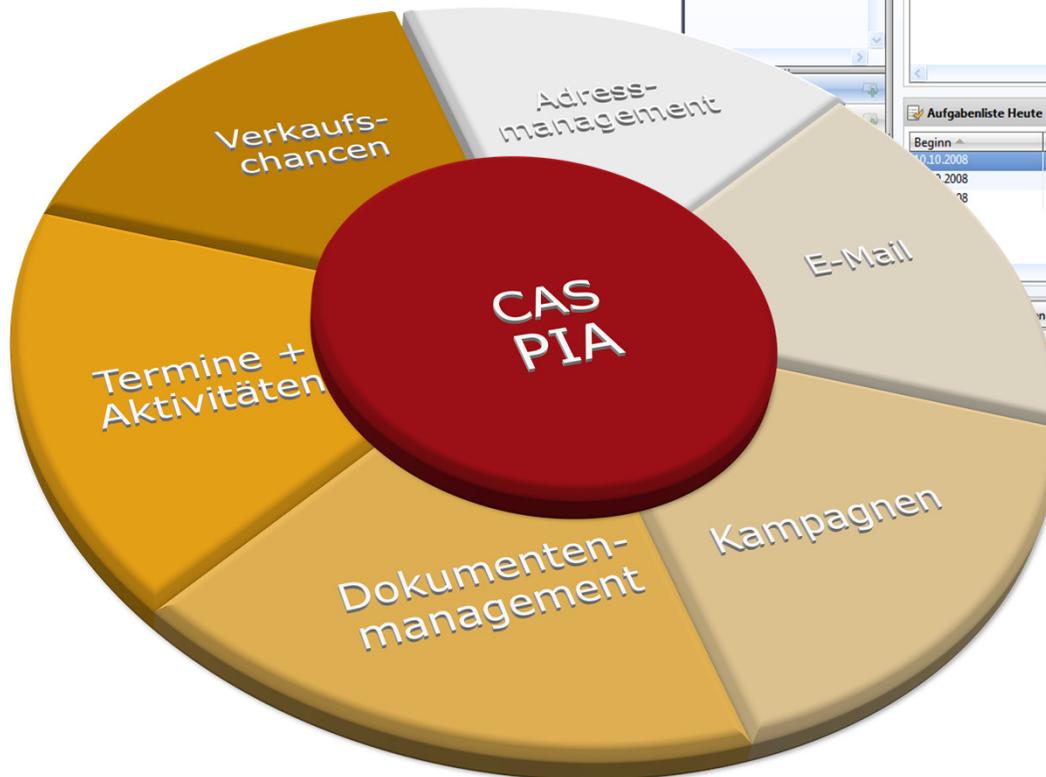
CAS PIA und Zielgruppe



- Die Zielgruppe ...
 - Kleinunternehmen und Selbständige (< 20 Mitarbeiter)
 - Unternehmen mit verteilten Arbeitsplätzen, Verbundgruppen, Verbände etc.
- ... und deren Herausforderungen
 - Kundenbeziehung wird maßgeblicher Wettbewerbsfaktor
 - Individuelle Kundenpflege benötigt hohe Automatisierung
 - Preis-Leistung ist zentrales Entscheidungskriterium
 - Kein IT-Knowhow, keine IT-Infrastruktur notwendig
 - Zentrale Datenablage



CAS PIA – das Produkt



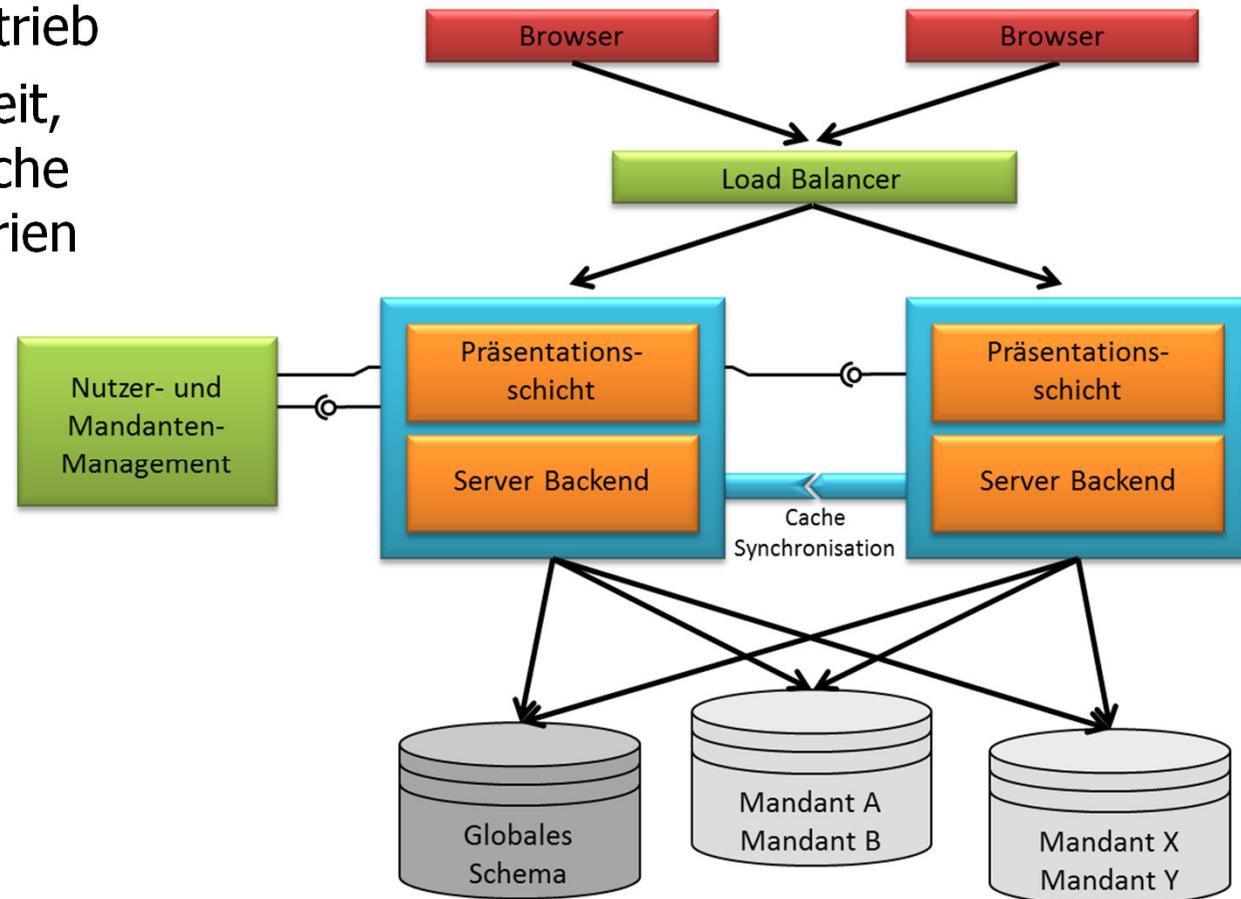
Erwartungen an SaaS-Produkte



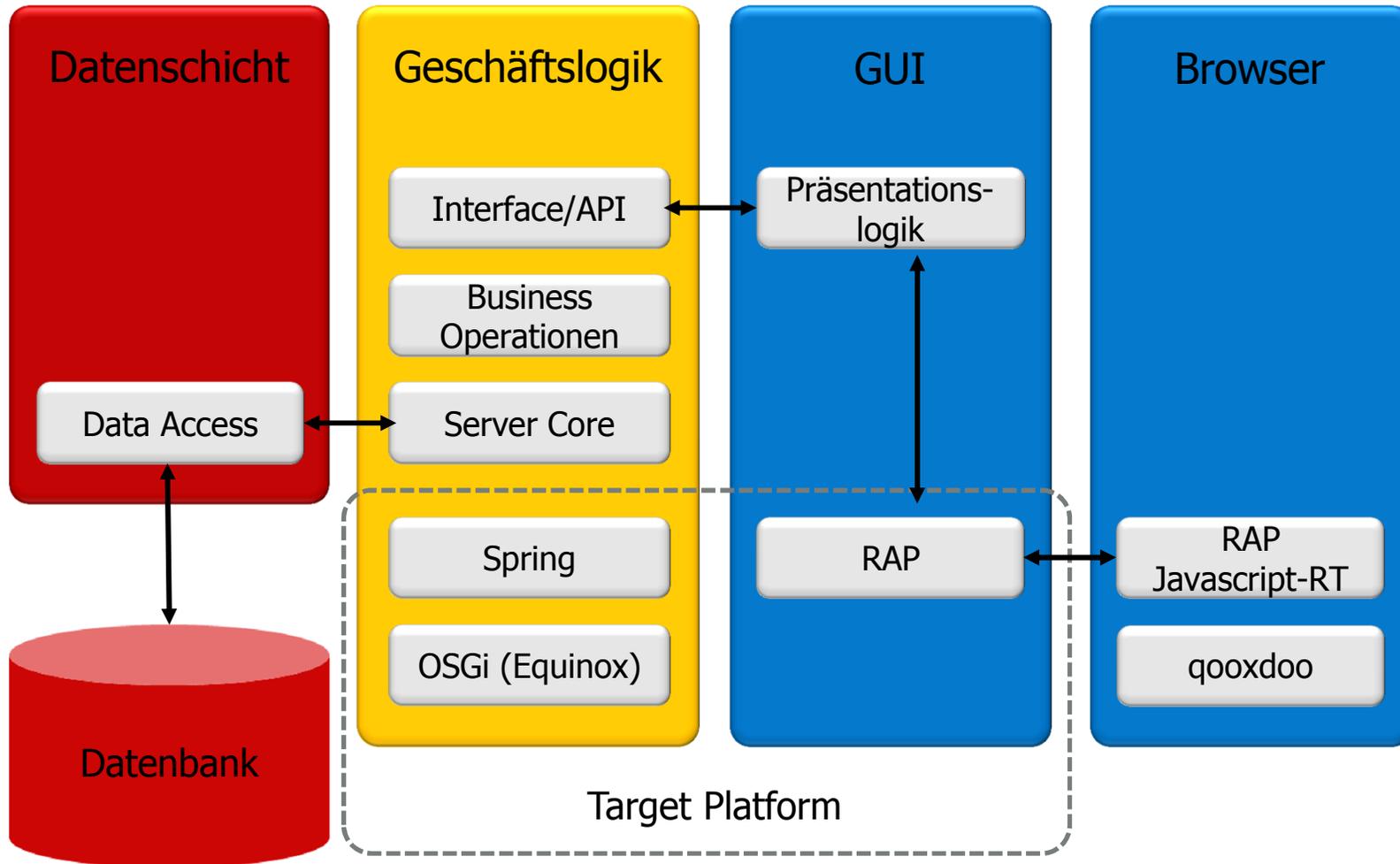
- Leichte Bedienbarkeit, Einstieg ohne Hürden
- Gute Anpassbarkeit an eigene Bedürfnisse
- Geringe Kosten
- Hohe Verfügbarkeit
- Gute Performance, Skalierbarkeit
- Sicherheit
- Gutes Zusammenspiel mit existierenden Lösungen

Technische Architektur

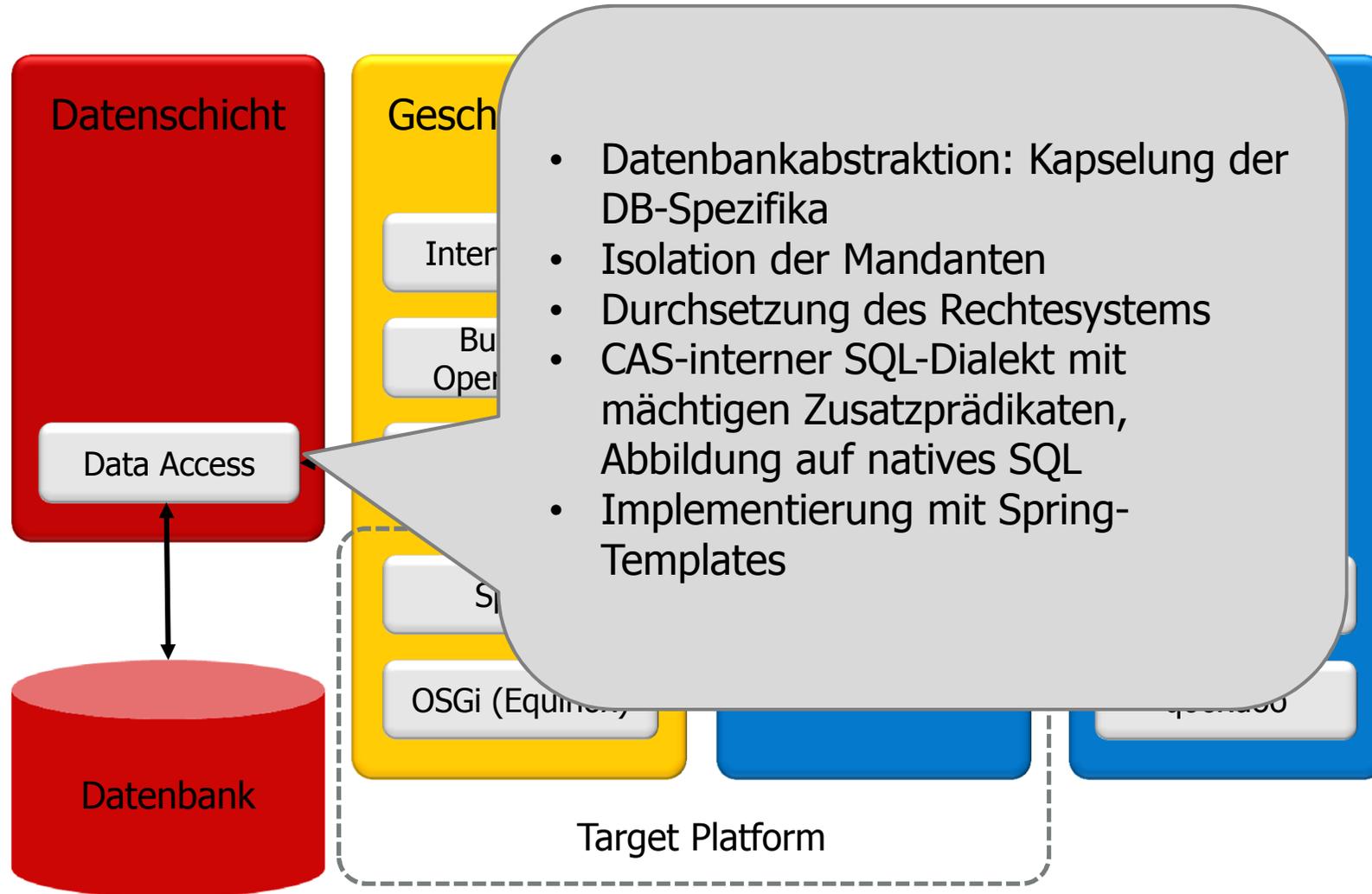
- Nutzung von Standardinfrastruktur, deploybar in VMs oder auf Standardservern
 - Günstiger Betrieb
 - Skalierbarkeit, unterschiedliche Einsatzszenarien



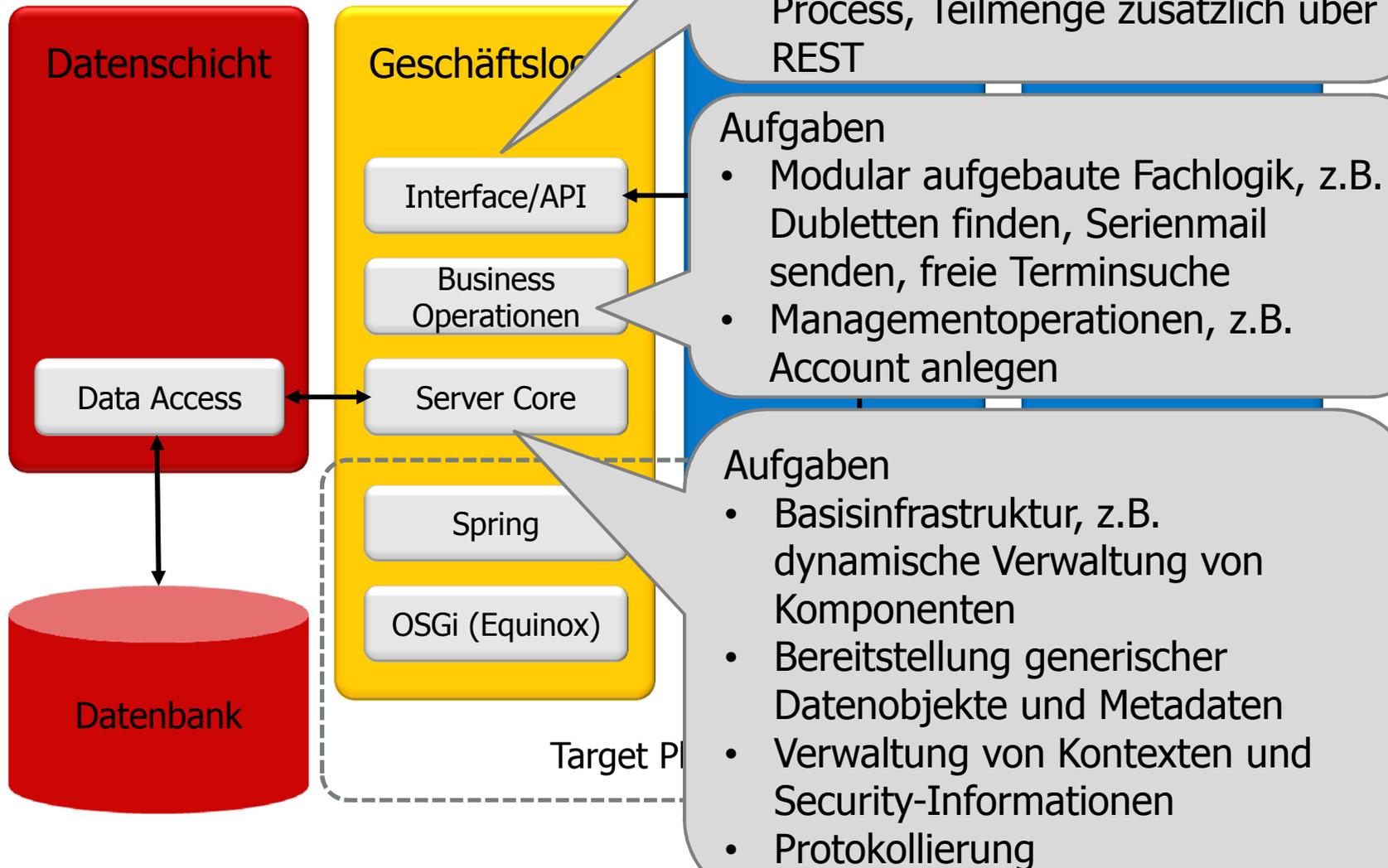
Logische Architektur



Logische Architektur



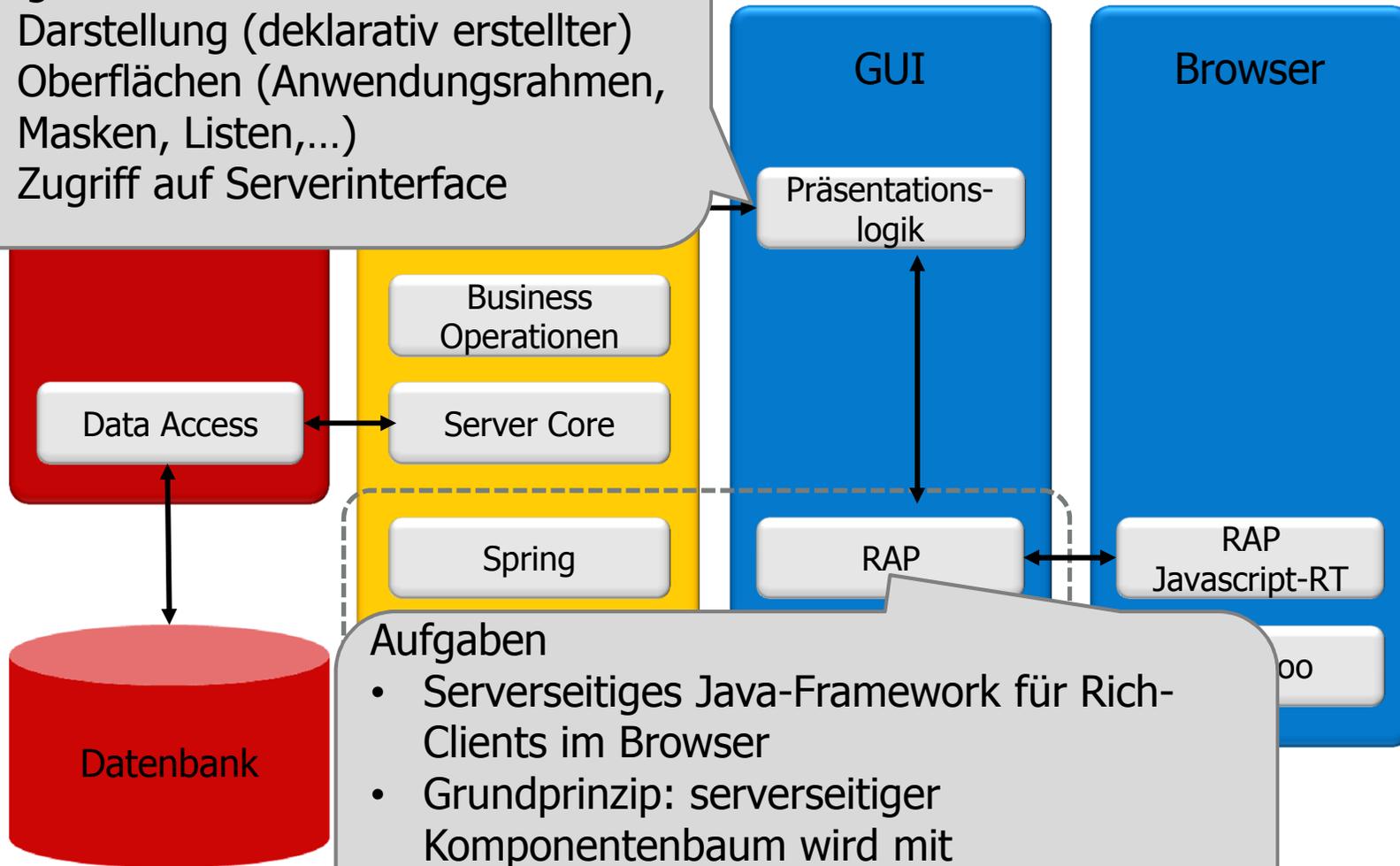
Logische Architektur



Logische Architektur

Aufgaben

- Darstellung (deklarativ erstellter) Oberflächen (Anwendungsrahmen, Masken, Listen,...)
- Zugriff auf Serverinterface



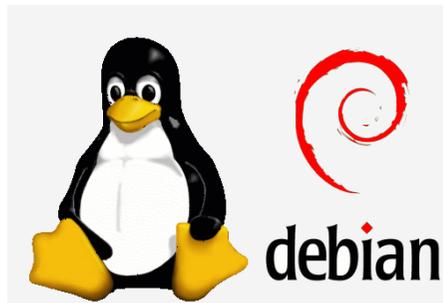
Aufgaben

- Serverseitiges Java-Framework für Rich-Clients im Browser
- Grundprinzip: serverseitiger Komponentenbaum wird mit browserseitigem über Lifecycle-Prinzip und XML-HTTP-Requests konsistent gehalten

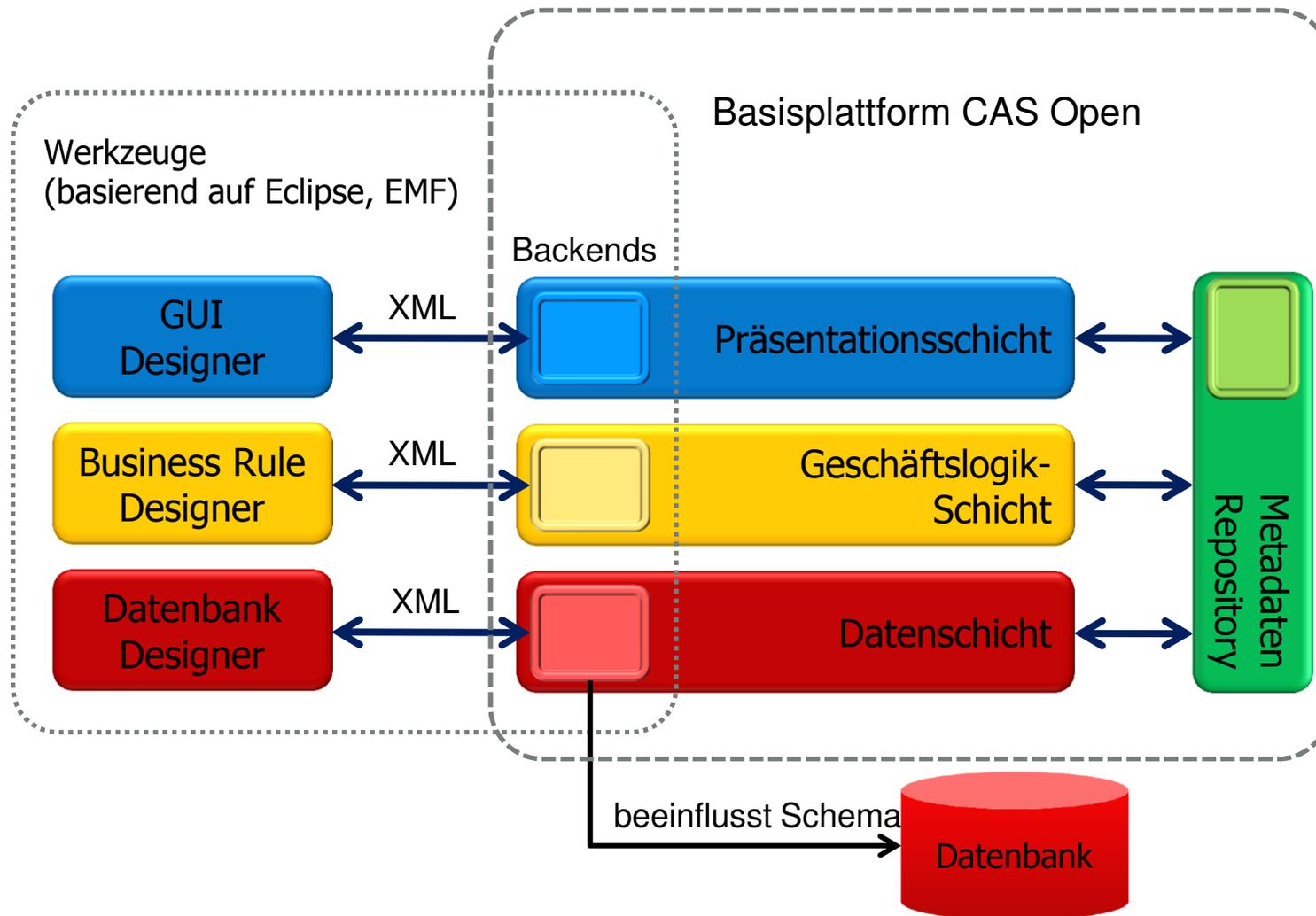
Technologien



- Linux-basierte Server mit Apache Tomcat und MySQL (mit Percona Patches), ggf. als VMs
- Spring, inkl. Dynamic Module Support (Blueprint), Spring-Security und Spring JDBC-Templates
- OSGi (Equinox)
- EH-Cache (inkl. Cache-Sync zwischen Servern)
- Eclipse Frameworks, insbesondere RAP, jFace und EMF

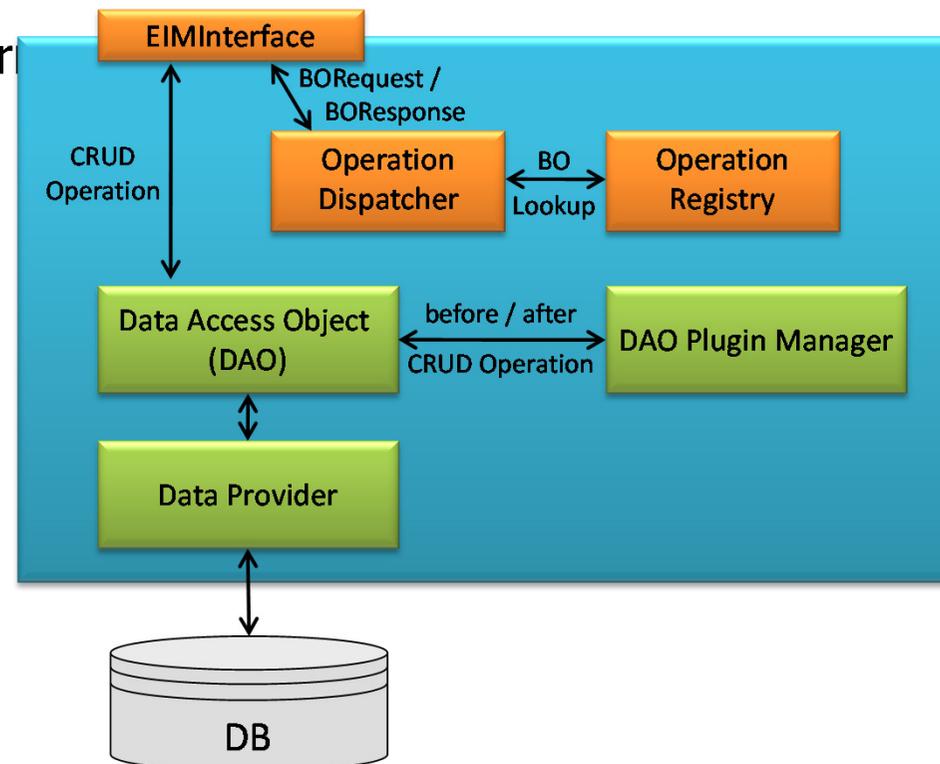


Anpassbarkeit durch Metadaten



Integrationsfähigkeit → API-Design

- Plattformgedanke:
 - Jede „wichtige“ Funktion als Webservice
 - Vorsehen von Anpassungs- und Erweiterungspunkten
 - Modulare Schnittstellen über generische Typen und durch Dispatcher/Registry ergänztes Command-Pattern
- Vorteile:
 - Konzentration der Logik im Backend
 - gute Konfigurierbarkeit, Ausbaubarkeit
 - Neue Clients (z.B. mobile Apps) können einfach entwickelt werden



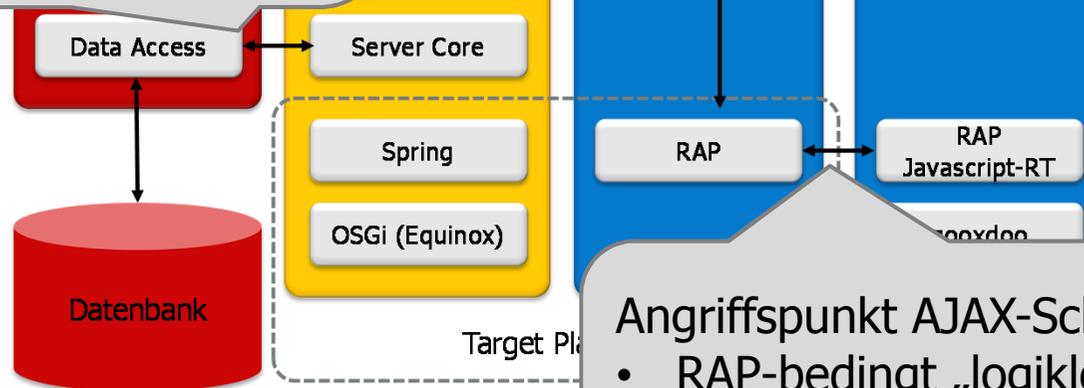
Sicherheit



- Teilsystemübergänge
- Isolation von Mandanten

Angriffspunkt DB-Schnittstelle

- Vermeidung von SQL-Injection durch überwachte, echte SQL-Übersetzung
- Erzwingen der Mandanten-Isolation



Angriffspunkt Service-Schnittstelle

- Operationen werden stets mit Benutzerkontext ausgeführt
- Integration von Ressourcenüberwachung

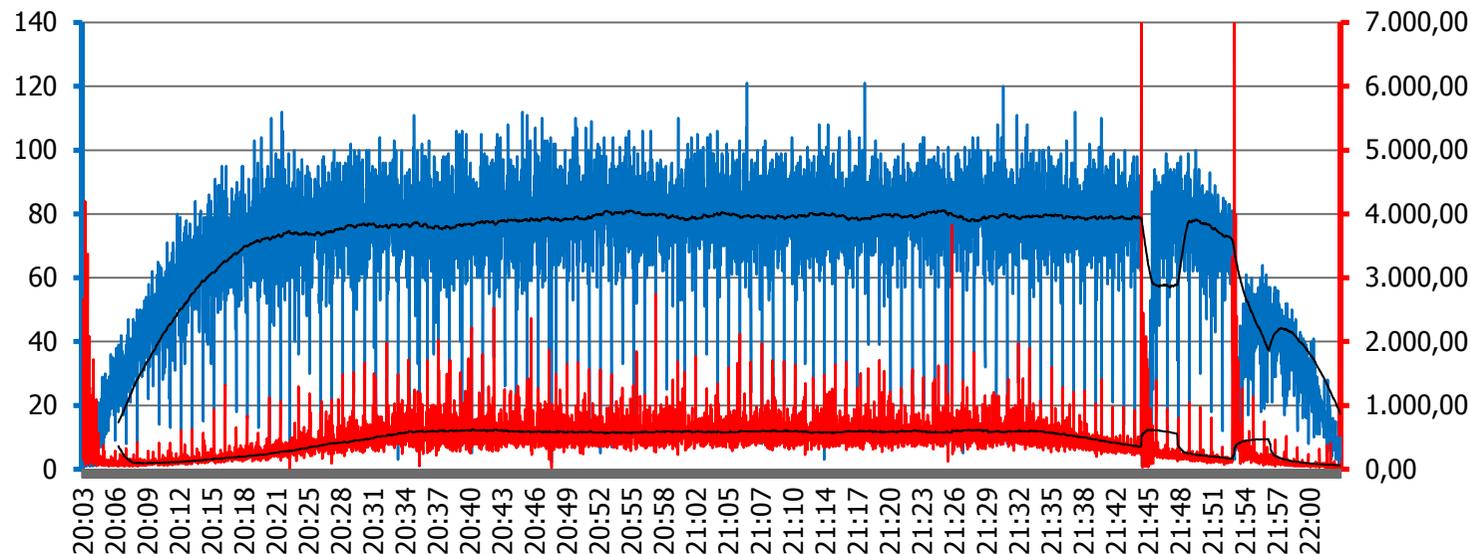
Angriffspunkt AJAX-Schnittstelle

- RAP-bedingt „logikloser“ Browser
- Sessionhijacking schwer aufgrund RAP-Mechanismen

Qualität



- Verschärfter Qualitätsbegriff:
 - kritische Fehler betreffen alle Kunden auf einen Schlag
 - Verhältnismäßig kleine „Performancekiller“ addieren sich schnell auf
- Maßnahmen:
 - Trimmen der Prozesse auf Qualität
 - Werkzeuggestützte Untersuchung, entwicklungsbegleitende Last- und Performance-Tests
 - Monitoring: „Sensoren“ für Standard-Lösung Nagios



Randaspekte



- Nutzerperspektive: Einstiegshürden vermeiden
- Anbieter muss sich auf neue Kundengruppe einstellen
 - Beratung durch Self-Service-Mechanismen ersetzen (Videos,...)
 - Billing
 - Support
- Trotzdem:
 - Kunden fordern persönlichen Kontakt!
 - Konventionelle Schulungen und Dienstleistungen bleiben gefragt

Zusammenfassung



- SaaS-Anwendungen – attraktiv durch Skaleneffekte für Anbieter und Kunden
- (offene) Standardtechnologien lassen sich gut dafür verwenden, erlauben gleichzeitig aber auch on-premise-Betrieb
- Skalierbarkeit über preisgünstige Standardkomponenten oder über (elastische) VMs möglich
- Metadaten-basierte Ansätze und dynamische Strukturen erleichtern Anpassbarkeit
- API-Bildung und Anpassungspunkte explizit vorsehen
- Sicherheit, Qualität (Robustheit) sind Kernherausforderungen für SW-Entwickler
- Zum Betrieb einer SaaS-Anwendung gehört mehr als nur deren Entwicklung!



Kontakt



Markus Bauer
CAS Software AG
E-Mail: markus.bauer@cas.de
www.cas-pia.de