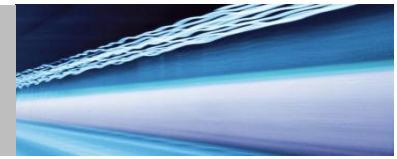


Die objektorientierte Hülle

Henning Schwentner

Jens Barthel



What's it all about?

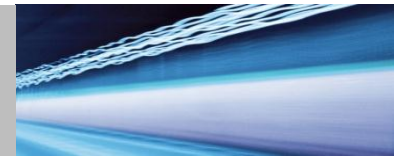
- Entwicklungserfahrungen aus der wundersamen Welt von SAP
- Objektrelationales Mapping zu Fuß in Legacy-Systemen
- Daraus resultierende agile Tendenzen

- Hier leider keine Raketenwissenschaft ☹️



Die Ausgangssituation

- Leasing-Branchenlösung *ERIKA*
- Monolithisches, historisch gewachsenes System
- Gemeinsame Datenbasis
- Geschrieben in ABAP
- Prozedural Imperativ implementiert
- Datenbank-Interaktion i.d.R. über spezielle Zugriffsfunktionen
- Keine automatisierten Tests



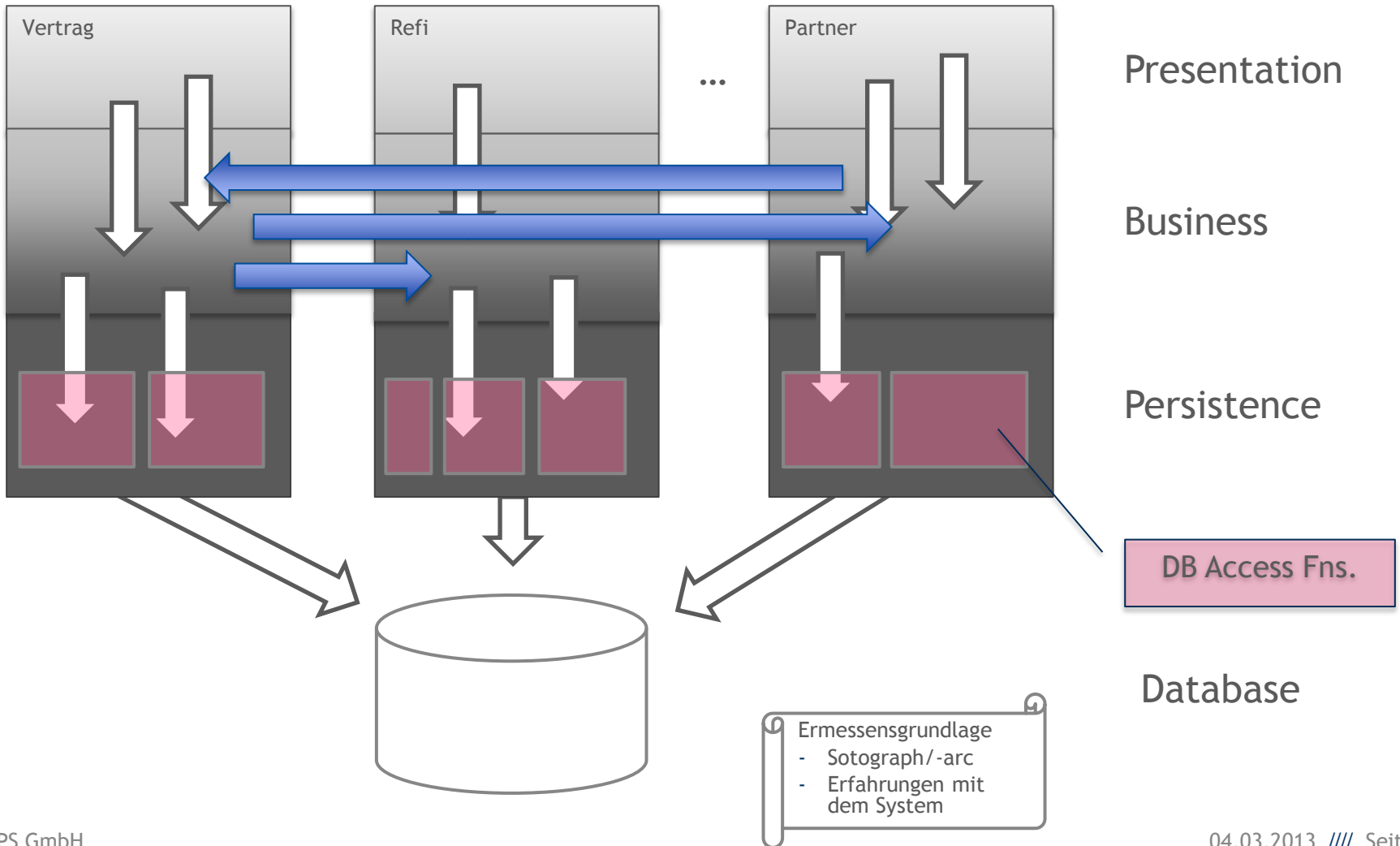


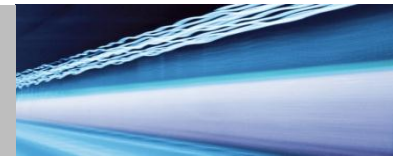
Die Aufgabe

- Anpassung, Wartung & Weiterentwicklung
- Unter Verwendung objektorientierter Programmiermechanismen
- Qualitätssicherung durch Unit-Testing
- Klare Trennung fachlicher Funktionen von der Datenbank

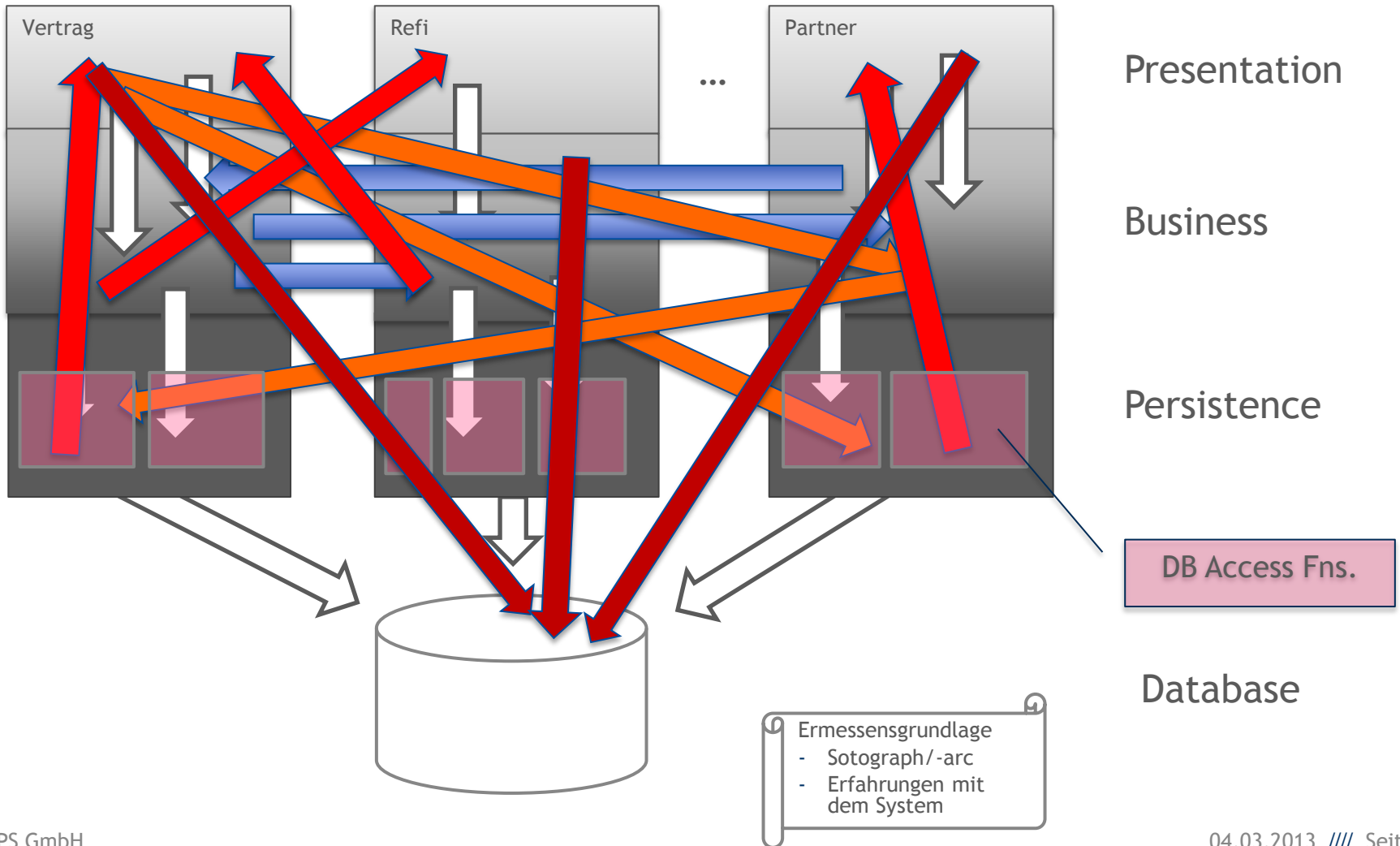


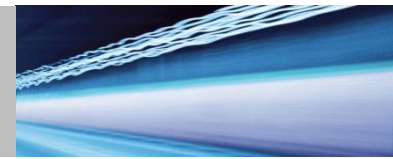
▪ Grober Überblick



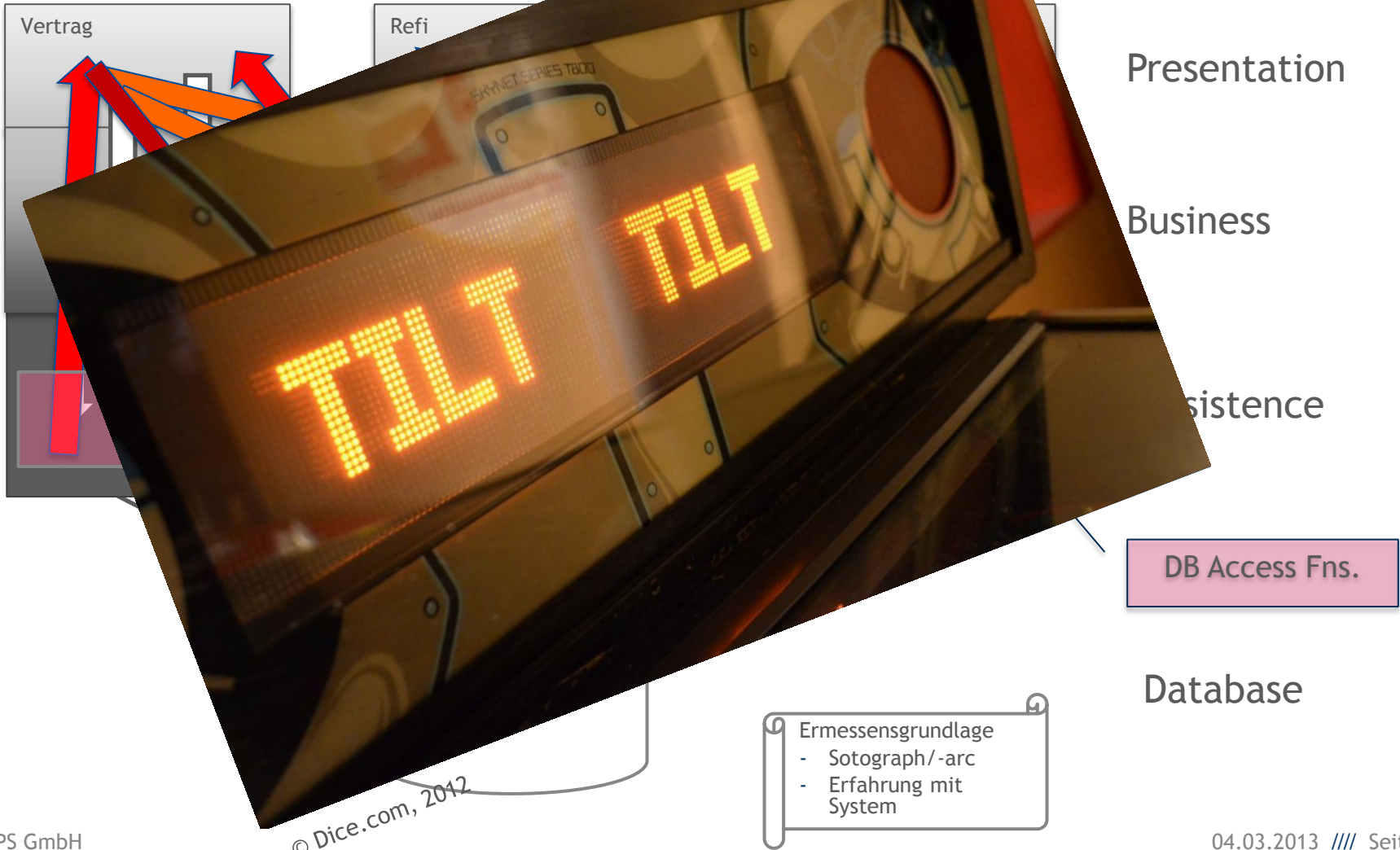


▪ Grober Überblick





■ Grober Überblick





Erste nüchterne Erkenntnis

Das System ist groß (~4.4 MLoC) und komplex.

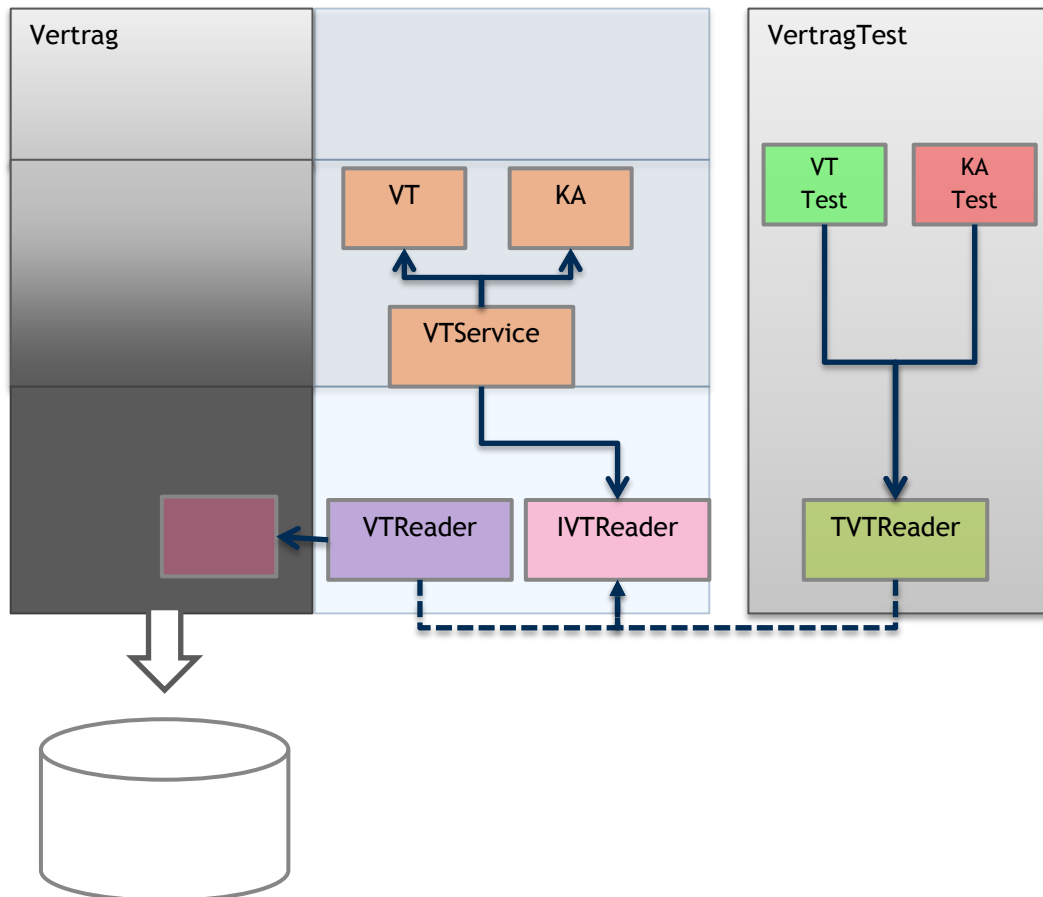
Eine Grundsanierung wäre teuer (>500 PT).

Es funktioniert dank etlicher Personenjahre manueller Tests.



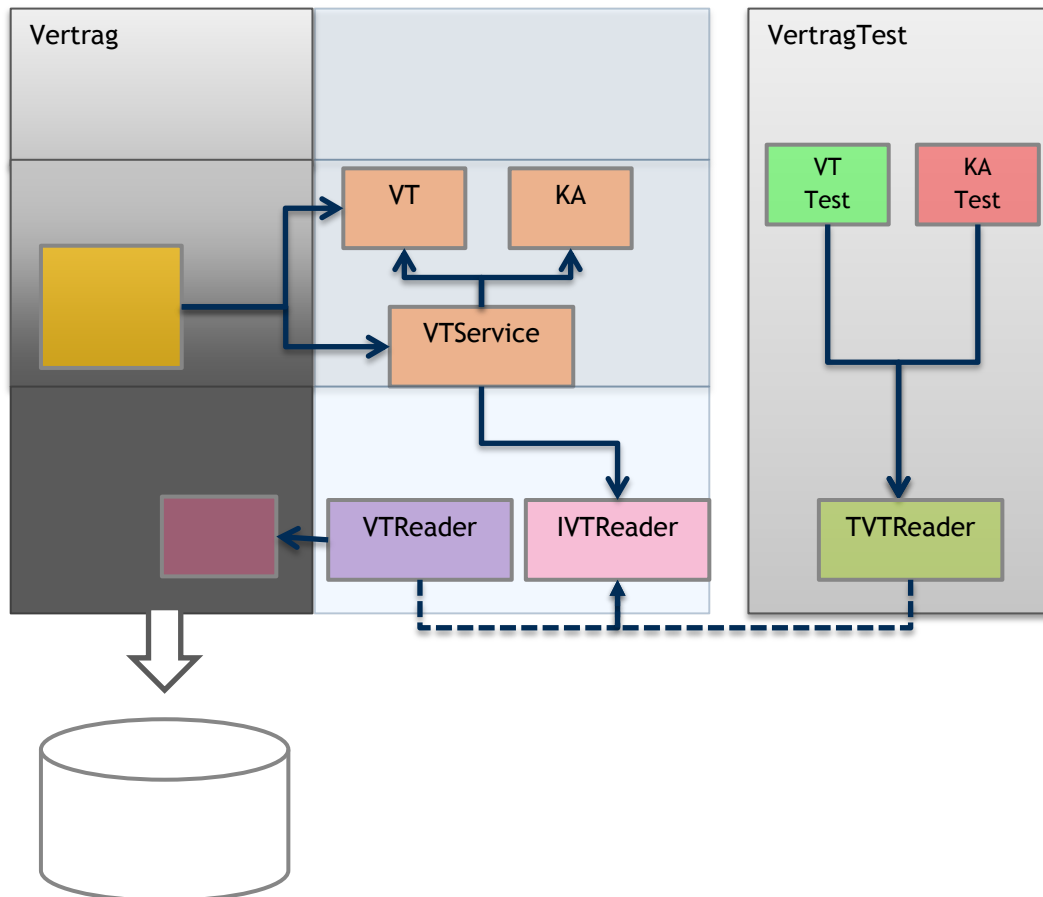
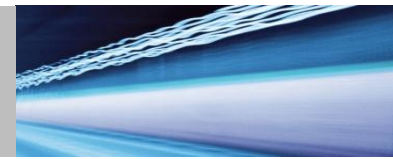
Was tun?

- O.g. Philosophie soll beibehalten werden
- Enthaltene Spezialwissen ist im Projekt abhanden gekommen
- System ist stark erodiert
- Big bang ist keine Option!
- Hohe Risiken durch fehlende Testunterstützung



Die Idee

- Spezifikation eines Datenbank-Mappings mit den objektorientierten Interfaces von ABAP
- Ihre Standard-Implementierungengreifen auf die o.g., imperativen Zugriffsfunktionen zu
- Objektorientierte Abbildung von Geschäftsobjekten
- Werden durch entsprechende Services verwaltet
- Mock-Implementationen ermöglichen Unit-Test



Die Idee

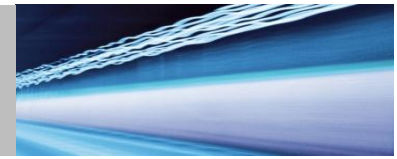
- Reader-Interfaces werden in die Geschäftsobjekt-Services injiziert
- Anpassungen und Erweiterungen im legacy code nutzen neue Zugriffsmechanismen
- Metaphern von Materialien und Automaten
- Nicht im Bild: MVC mit Legacy-GUI-Komponenten



- DB-Reader & Testing
 - Minimalinvasiv
 - Wachsende agile ‚green zone‘
 - Code-Qualität steigt!
 - Have your cake and eat it too. Tests können nach Belieben auf persistenten Daten, oder Mocks ausgeführt werden.
 - Entkopplung ermöglicht Unit-Testing auf unterschiedlichen Systeminstanzen
 - Problem: Unit-Tests können zu langen Fingerübungen werden. Im Zweifelsfall müssen viele Mocks initialisiert werden.
 - Testdaten im Dateisystem
 - Entwickler bekommen eigene Datenbanken
 - Was würden Sie vorschlagen? Nota bene: Nur ein SAP Mandant für Entwickler



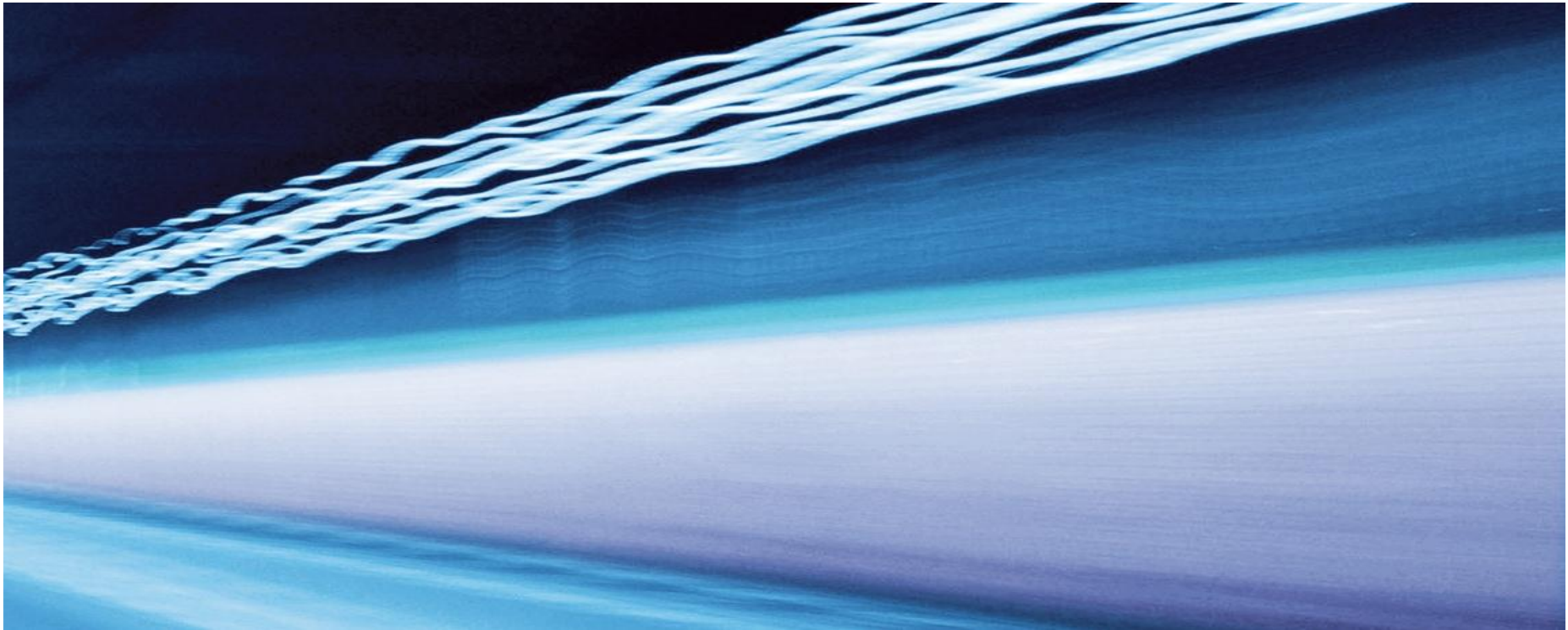
- Geschäftsobjekte & Services
 - technisch unzusammenhängende Datentypen werden fachlich geklammert
 - Deutliche Reduktion an boilerplate code um nötige Daten nachzuladen. Ein Serviceaufruf genügt
 - Risiko: Performance-Einbußen im Batch durch internen Aufbau von Geschäftsobjekten
 - DB-Reader mit Caching-Unterstützung.
 - Höher angesiedelte Sperrkonzepte im Legacy-Code unterbinden konkurrierende Updates weitestgehend.
 - Batch läuft nachts ... meistens
 - Lazy loading von Teilgeschäftsobjekten



Zweite nüchterne Erkenntnis

Der zentralisierte Datenbank-Zugriff des Altsystems ist für das Verfahren fundamental wichtig.

Die zugrundeliegende Programmiersprache muss Polymorphie unterstützen.



Vielen Dank!

Was denken Sie?

henning.schwentner@c1-wps.de

jens.barthel@c1wps.de