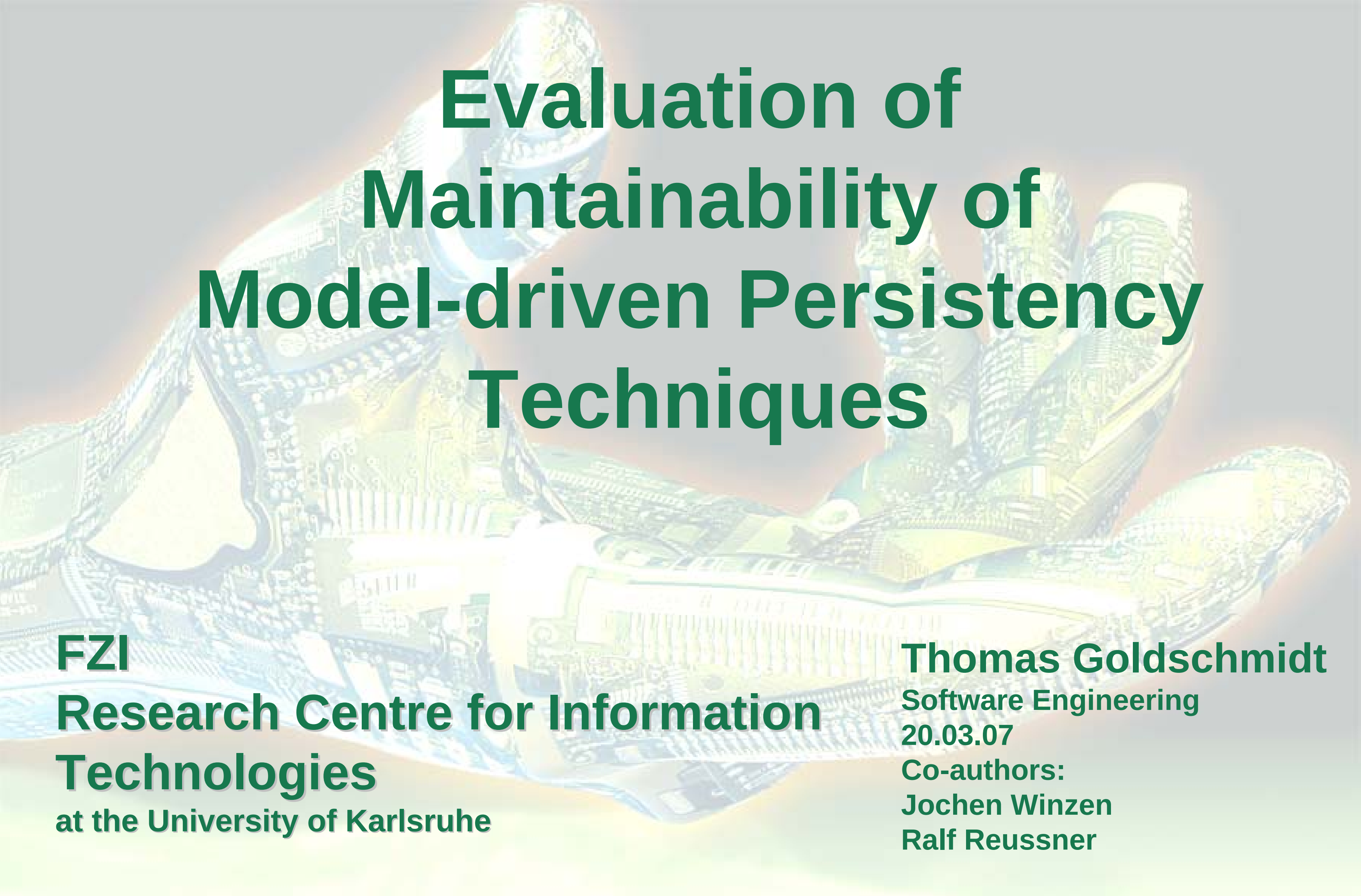


Evaluation of Maintainability of Model-driven Persistency Techniques

FZI
Research Centre for Information Technologies
at the University of Karlsruhe

Thomas Goldschmidt
Software Engineering
20.03.07
Co-authors:
Jochen Winzen
Ralf Reussner

0. Agenda

1. Motivation
2. Evaluation Testbed
3. Derivation of Metrics
4. Assessed Persistency Techniques
5. Current Status and Preliminary Results
6. Questions

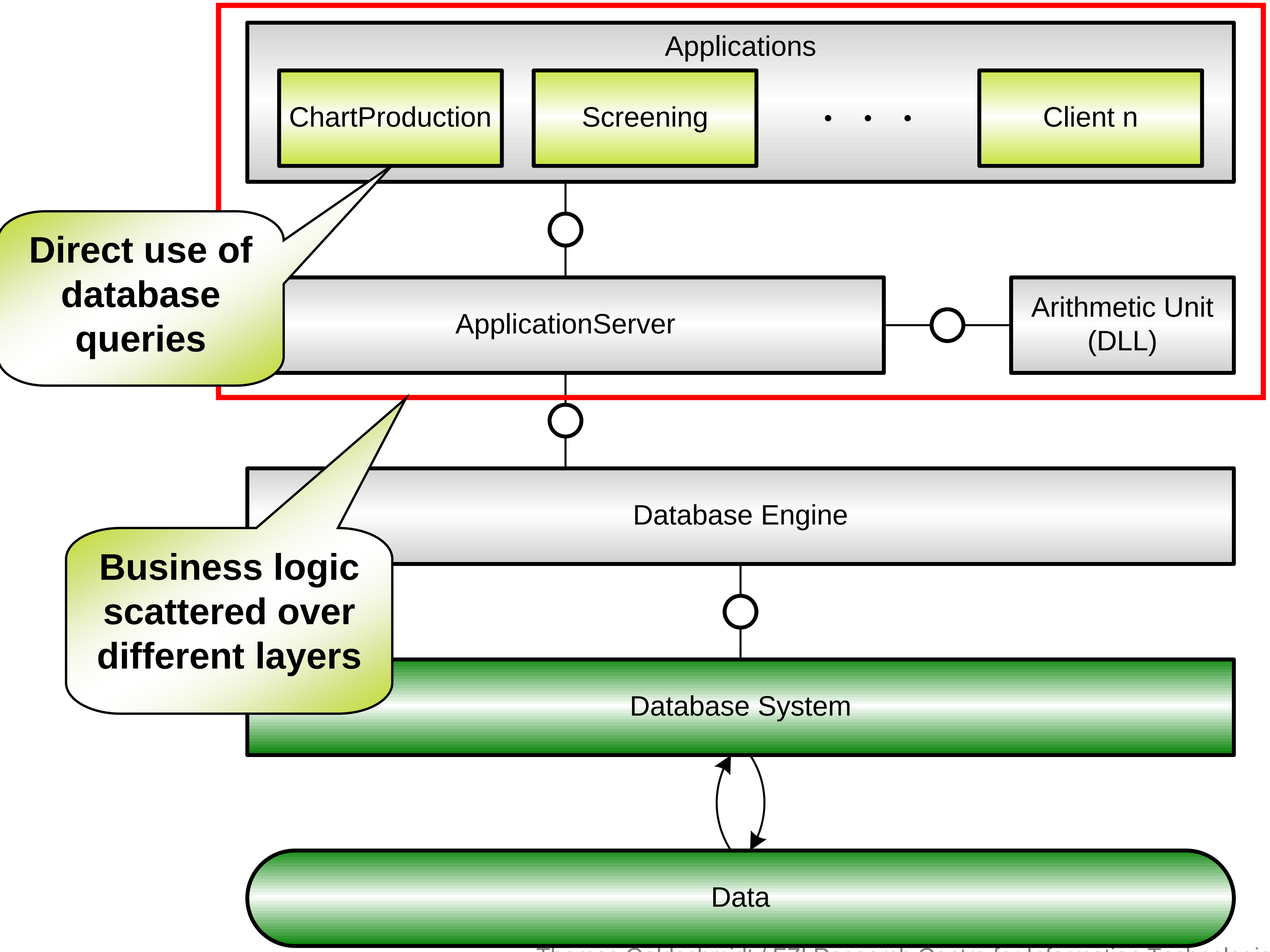
1. Motivation

- Systematic evaluation of the benefits and drawback of model-driven approaches
- Focus on:
  - Area of adapter generation to map object oriented models to relational databases
  - Migration and modernization of legacy systems
- Evaluation criteria
  - Maintainability
  - Performance
- Goal
  - Derive a catalogue of best-practices and anti-patterns
  - Guidelines for design decisions

2. Testbed

- Developed based on a real-world legacy application:
  - MESCOR, extensive program suite for financial research
  - Group of standalone applications using common middle/database tier
  - Different applications with different requirements to persistence layer
    - Read intensive
    - Complex read/write operations
    - User interaction scenarios (short response time required)
  - Database with more than 100 tables
  - Written in Borland Delphi
  - Communicates via DCOM and sockets

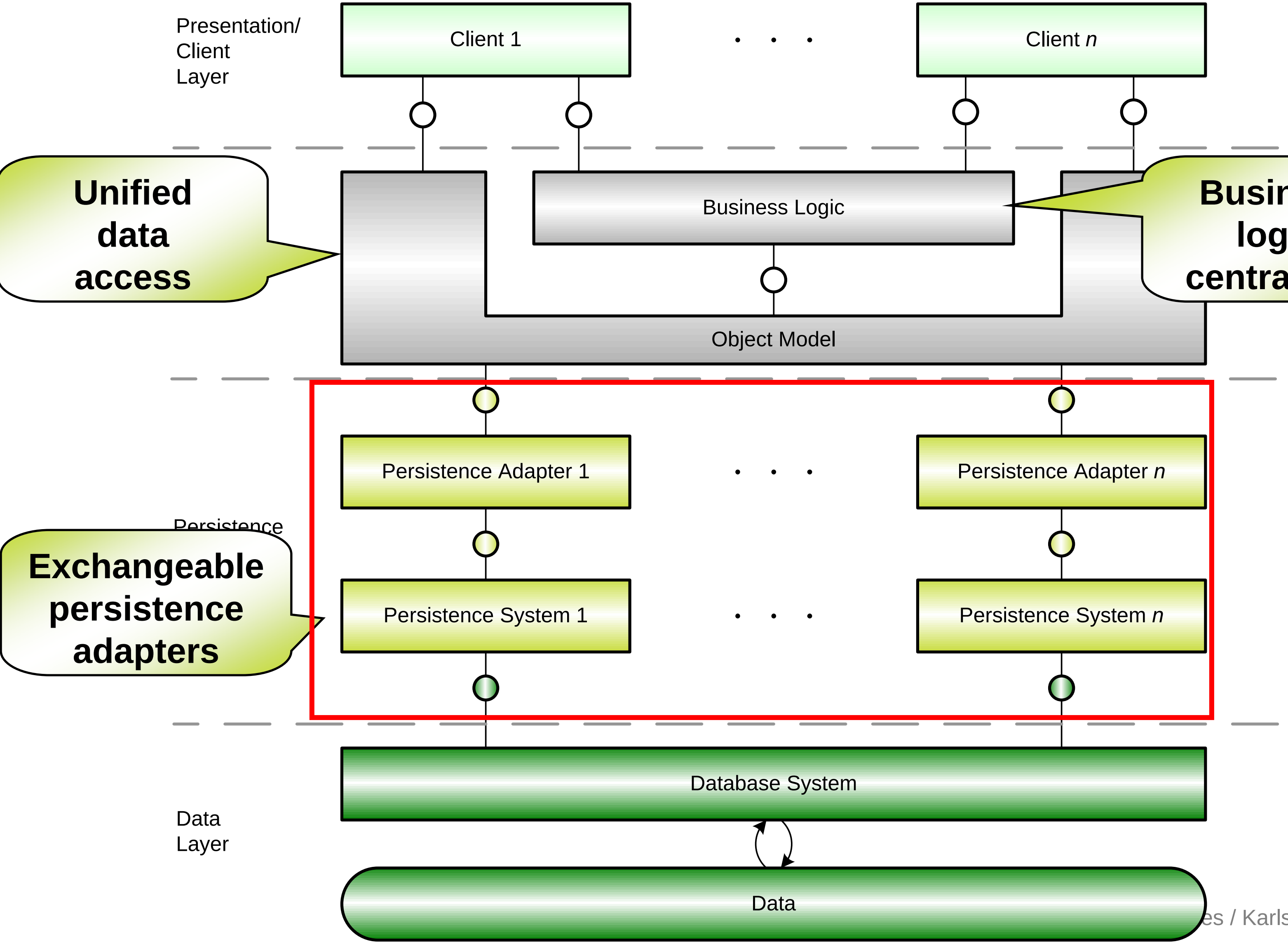
2. Testbed - Legacy System



Direct use of database queries

Business logic scattered over different layers

2. Testbed Architecture



Unified data access

Business logic centralized

Exchangeable persistence adapters

3. Derivation of Metrics

- Defined using the Goal Question Metrics (GQM) Approach [Basili,1994]

1. Define goals, what should be achieved by the measures?

   In our case comparison of:

   - Maintenance
   - Performance

2. Ask Questions on how the comparison can be achieved
   - E.g.: What is the impact of the chosen persistence adapter on the start-up time of the application?

3. Find metrics that help to answer the questions
   - Time, memory usage, LOC etc.

3. Derivation of Metrics – Sample GQM for Maintainability

- **Goal 1:**
  - **Purpose:** Comparison
  - **Object:** Different persistency techniques
  - **Issue:** Maintainability
  - **Viewpoints:** The software development and maintenance team

- **Question 1.3:** How big is the effort to extend the persistency layer with a new persistent class?

- **Metrics:**
  - **M1.3.1:** Time to conduct the change in *hours*.
  - **M1.3.2:** *Amount of files and/or models* that need to be touched.
  - **M1.3.3:** *Amount of test and debug runs* that were neededto pass all tests after the change.

3. Derivation of Metrics

- Goals for Maintainability and Performance
- 10 Questions
- 26 Metrics
- Measurements carried out in 3 different scenarios

1. Mapping the newly created object model to the legacy database
2. New database schema according to the object model (e.g. directly generated)
3. Migrating the mapping from the legacy database to a newly defined database schema.

4. Assessed Persistency Techniques

- Manually implemented persistence layer
  - ADO.NET
- Persistence frameworks
  - NHibernate
  - OpenAccess
- Generated adapters
  - Delta Software Legacy Integration Tool Suite
  - i3 Design MDRAD
- Combination of generator and persistence framework
  - Out-of-the-box generators
    - AndroMDA NHiberante Cartridge
  - Project specific generators
    - Developed with Delta Software‘s HyperSenses (with Pattern by Example support)
    - Developed with OpenArchitectureWare
    - Developed with Interactive Objects ArcStyler

5. Current Status and Preliminary Results

- Implemented first application „Chartproduction“ for legacy database scenario using:
  - NHibernate (9 days)
  - ADO.NET (12 days)
  - Currently developing OpenAccess mapping
- Implemented generator for NHibernate mappings + supporting classes with:
  - Delta Software HyperSenses (3 days)
  - Interactive Objects ArcStyler (3.5 days)
  - Next step: OpenArchitectureWare
- First performance measurements: Batch run of “Chartproduction” application
  - Revealed slight differences between NHibernate and ADO.NET depending on the scenario

6. Questions

Questions?

Thank You!