Laura:
The Xerox Advanced Print Server Architecture

THE DOCUMENT COMPANY
XEROX

Version I

Contributors:

Markus Doessegger, Jean–Christophe Godinaud, Egbert Hess, Thomas Jenny, Tiziano Leidi, René Lanz, Roland Meier, Pascal Pagny, Markus Rohner, Thomas Schumacher, Pierre Staquet, Markus Steiner, Severin Stoeckli, Vincent Tscherter

Laura:
The Xerox Advanced Print Server Architecture

THE DOCUMENT COMPANY
XEROX

- Einführung
- Entwicklungsumgebung
- Implementation
- RMI
- GUI
- Speicherüberwachung
- Schlussfolgerungen und Fragen

Introduction

Laura Requirements

THE DOCUMENT COMPANY
XEROX

Photocopiers
Printers

Shell Scripts

Print Server
C

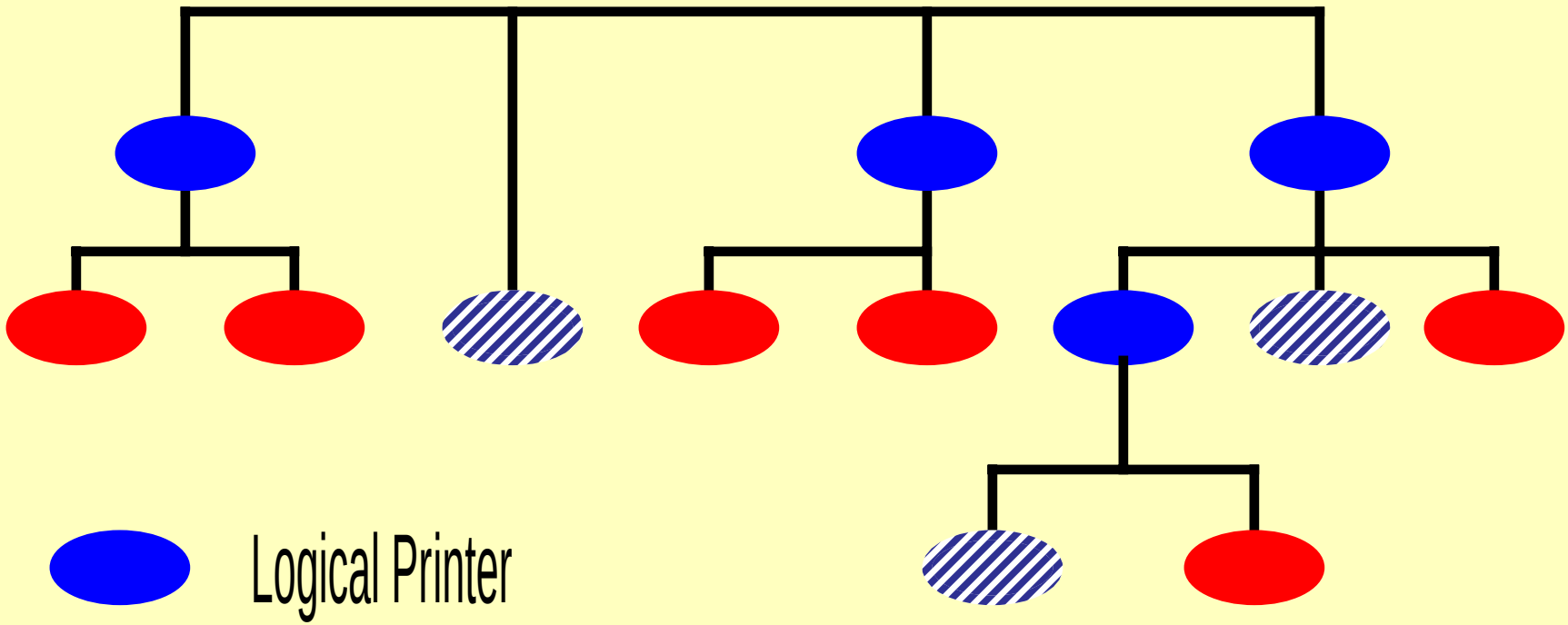
Laura

Requirements

- Flexible and Open
- Modular and Extendable
- Easy to Maintain
- Portable

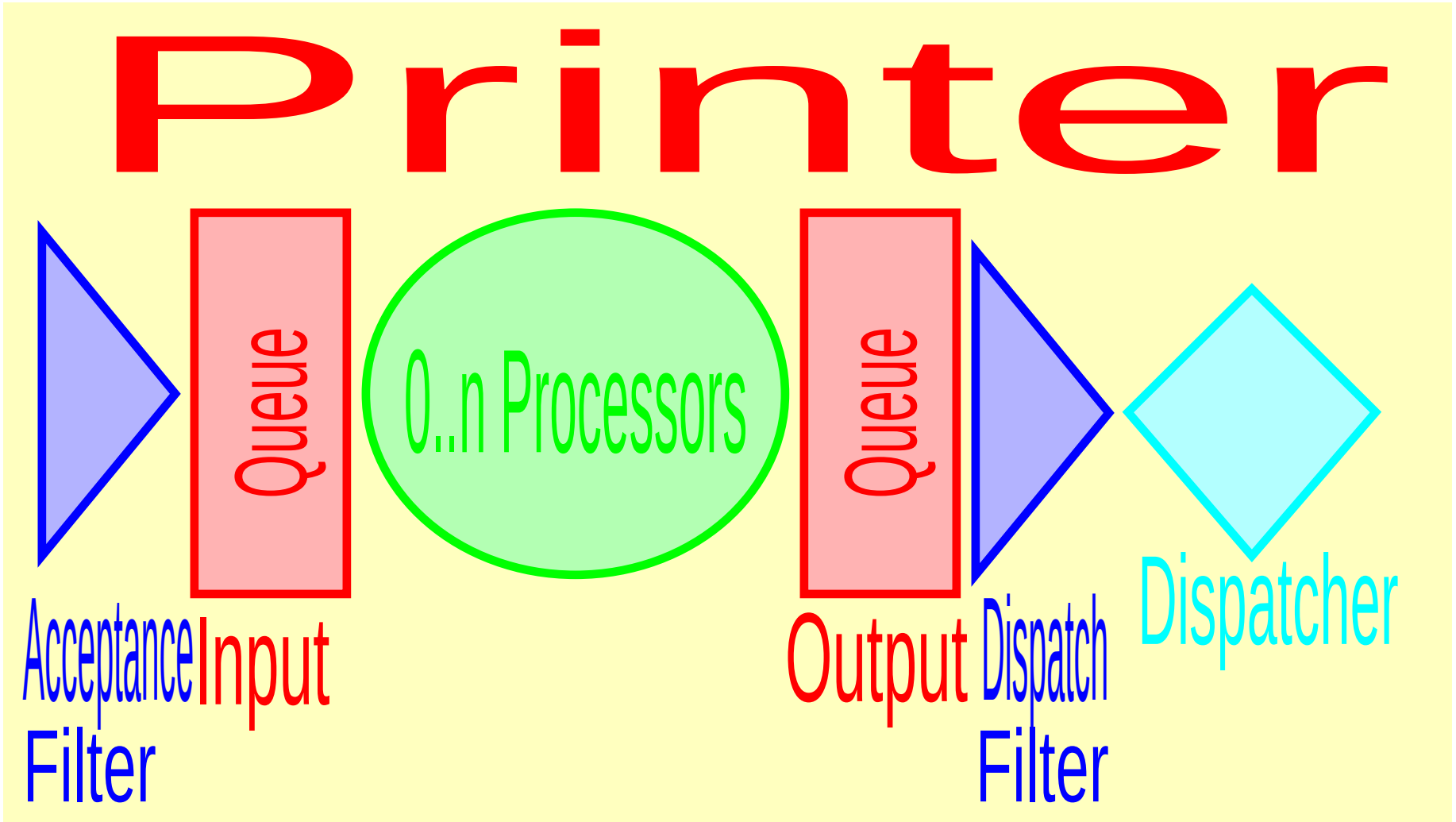
Introduction

Automated Print Service



Introduction

Printer



Laura
IntelligentOutput Architecture

THE DOCUMENT COMPANY
XEROX

Object Attributes

Printer

printer-state=idle

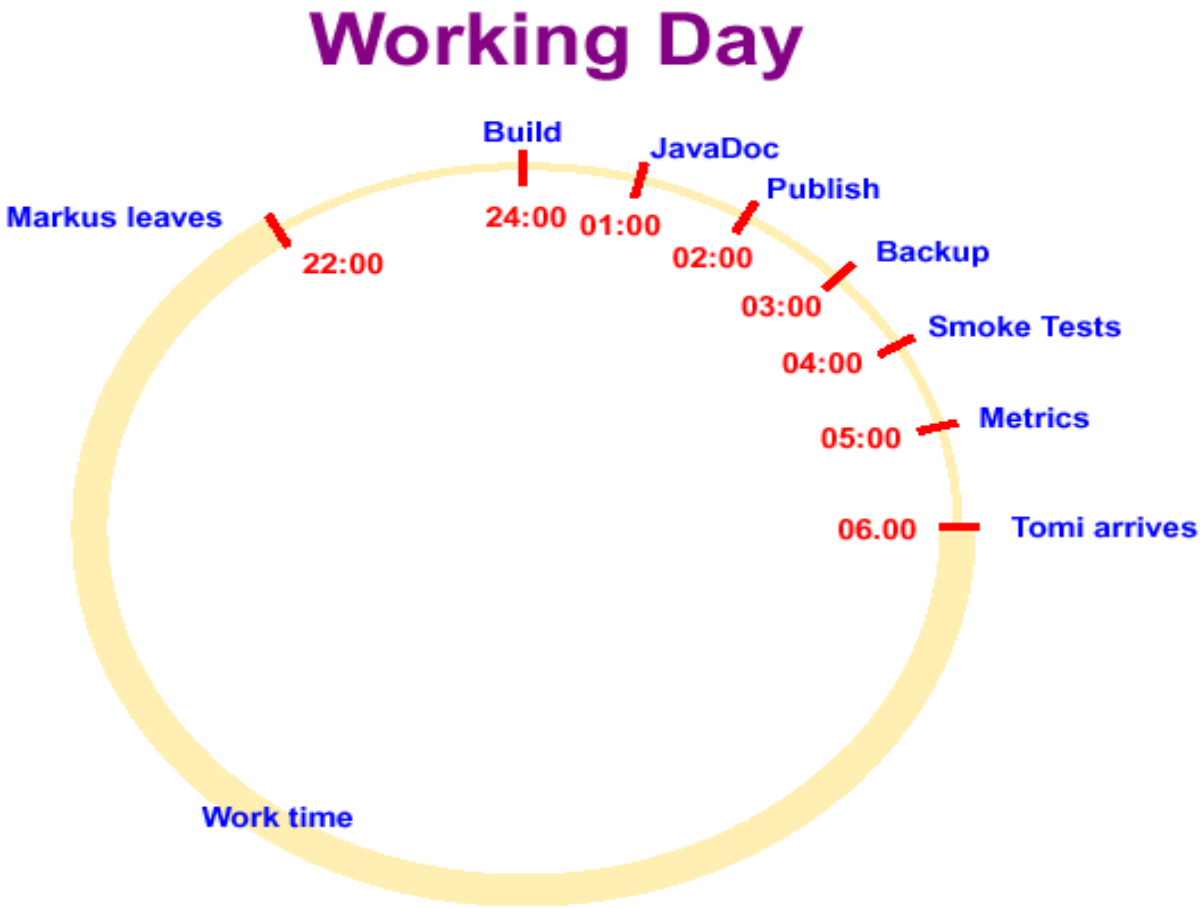
media-ready=iso-A4-white

auxiliary-sheets

job
document
medium
server

Development environment

Der 24-Stunden-Kreis



- Daily Build
- Daily Smoke
- Daily Publish

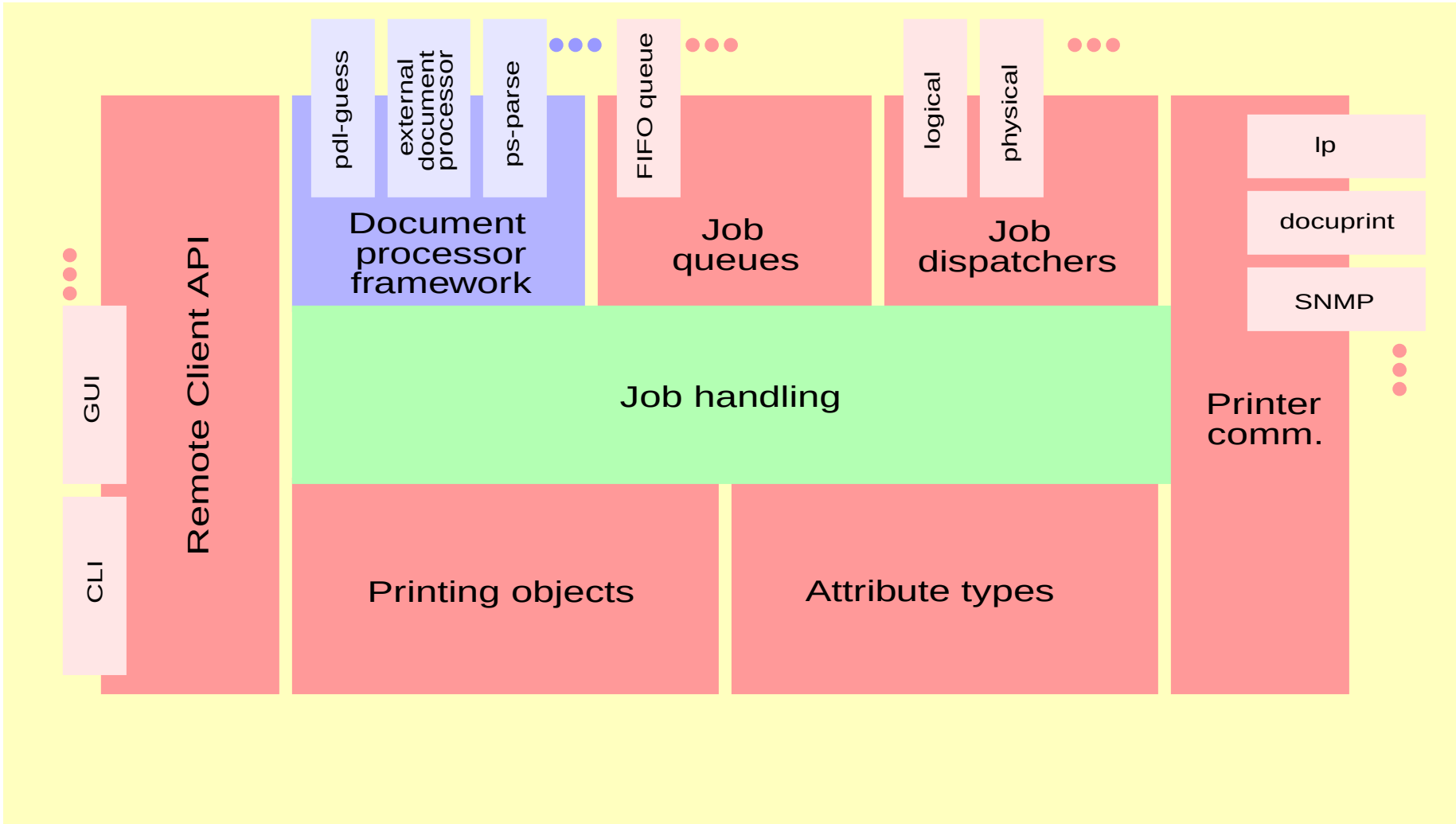
Development environment

Tools

THE DOCUMENT COMPANY XEROX

- **SNiFF+ 3.2 von Takefive Software**
  RCS
  GNU make
- **TogetherJ von TogetherSoft**
- **ObjectStore von ODI**
- antlr (Parser-Generator)
- javadoc (Dokumentations-Extraktor)
- junit (Test-Engine)
- Shell Scripts (Tests, Installation, Konfiguration, ...)

Architektur



Themen


Authentifizierung/
Autorisierung
flexible Filter (antlr)
Drucker-Ansteuerung
Drucker-Eigenschaften
erkennen/nutzen
Printing Objects mit Attributes
Memory Management
Events
Datenbank-Anbindung
skalierbares GUI ("Echtzeit";
>500 Objekte in Tabelle)
Document
Processing
Job Handling Engine (Pausieren/
Wiederaufnehmen von
JobProcessing)
Postscript (Parsing;
"Coding" (DPI))
RMI
Media-Unterstützung

Themen


Authentifizierung/
Autorisierung
flexible Filter (antlr)
Drucker-Ansteuerung
Drucker-Eigenschaften
erkennen/nutzen
Printing Objects mit Attributes
Memory Management
Events
Datenbank-Anbindung
skalierbares GUI ("Echtzeit";
>500 Objekte in Tabelle)
Document
Processing
Job Handling Engine (Pausieren/
Wiederaufnehmen von
JobProcessing)
Postscript (Parsing;
"Coding" (DPI))
RMI
Media-Unterstützung

RMI

THE DOCUMENT COMPANY
XEROX

RMI ist simpel, einfach

Nötig:

- ein remote interface

```
public interface RemotePrintingObject extends Remote {
  public Attribute getAttribute( String attrName )
    throws RemoteException;
  public void setAttribute( Attribute attr )
    throws remoteException;
  public void delete()
    throws RemoteException;
}
```

- eine Implementationsklasse für dieses Interface
- einen Client, der das Remote interface benutzt

=> Voila: läuft !

RMI-Probleme

THE DOCUMENT COMPANY XEROX

- RMI call ist teuer (Skalierbarkeit)
- Explosion der Remote Interfaces
- Memory-Kontrolle im Server

=> RMI ist einfach, verteilte Systeme sind es nicht

Verbessertes Remote Interface

THE DOCUMENT COMPANY XEROX

- weniger Interfaces, der Server hat grössere Kontrolle
- mächtigeres Interface (gibt pro RMI-Call mehr Daten zurück)
- kein direkter Zugriff für Clients auf die PrintingObjects mehr, nur noch über Vermittler:

```
public interface RemotePOwarehouse extends Remote {
  Values[] getAttributes( String[] objectIds,
                          String[] attrNames[] )...;
  void setAttributes( String objectId, Values[] values )..;
  void perform( Operation op, String[] objecIds ) ...;
}
```

Graphical User Interface

Übersicht

THE DOCUMENT COMPANY XEROX

Live view:

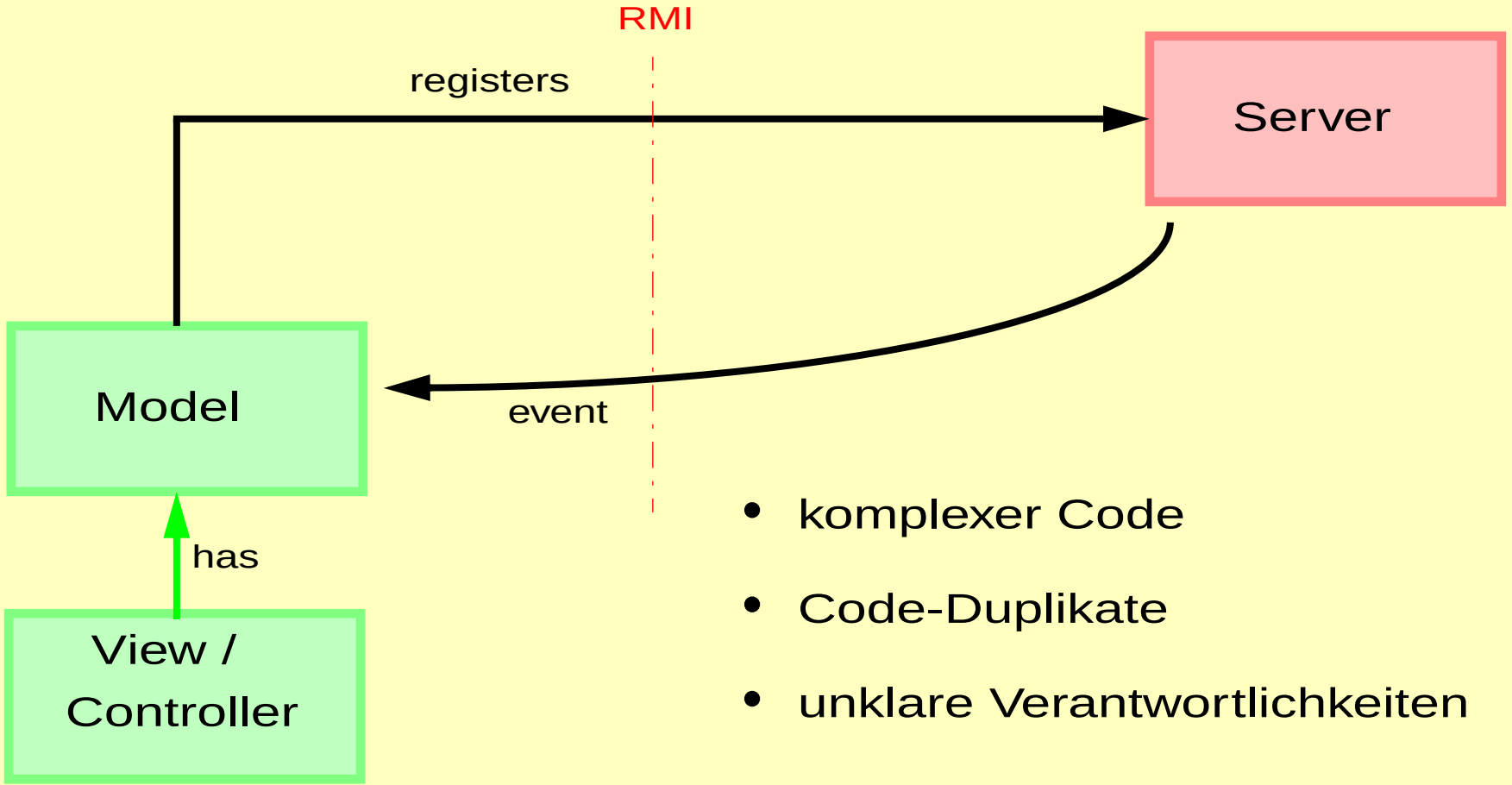
- dynamische Sicht
- aktuelle Zustände und Zustandsänderungen
- laufend aktualisierte Sortierung

Snapshot view:

- statische Sicht
- von Zustandsänderungen unabhängig
- Attribute editieren

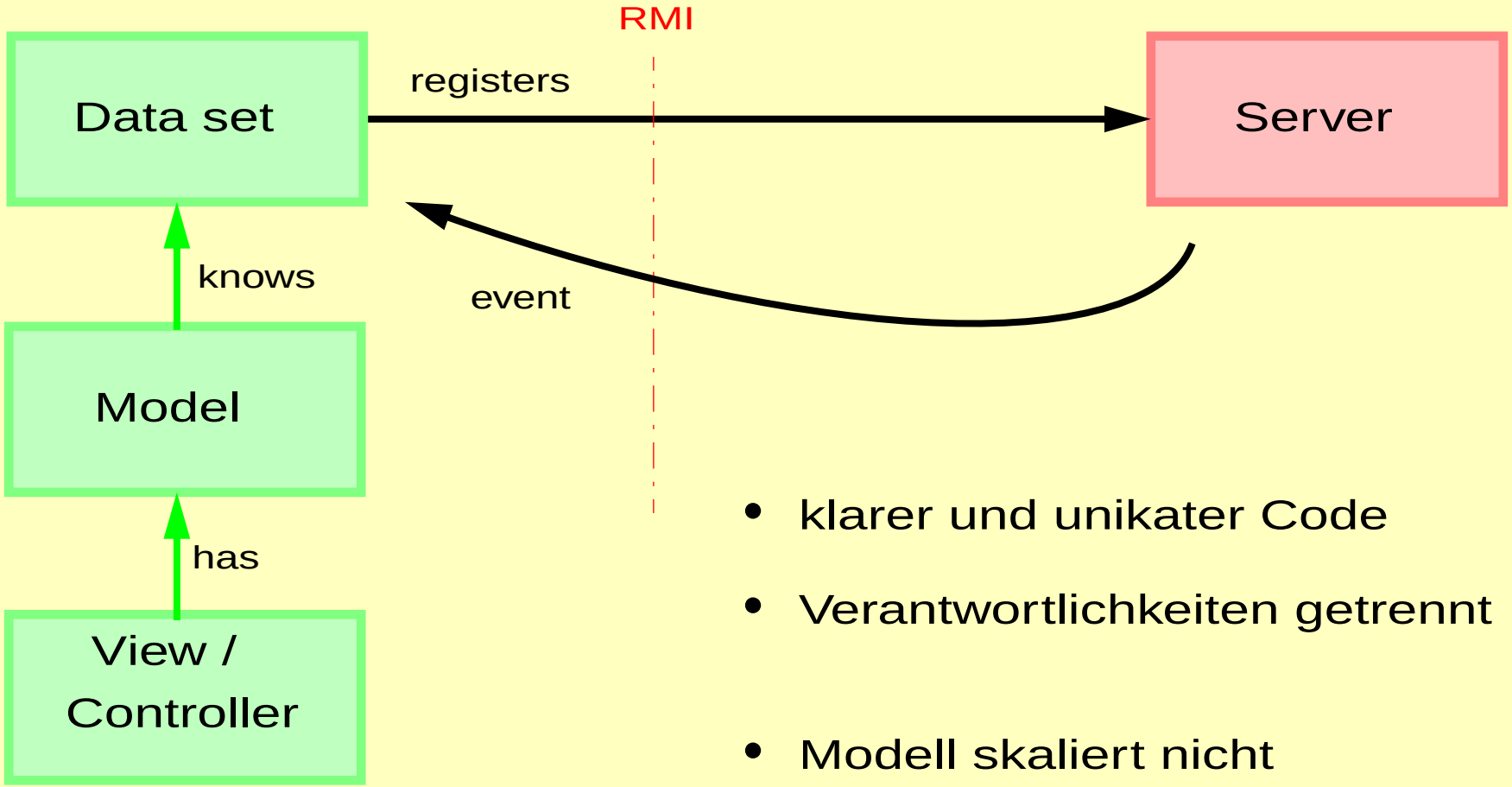
Graphical User Interface

1. MVC.



Graphical User Interface

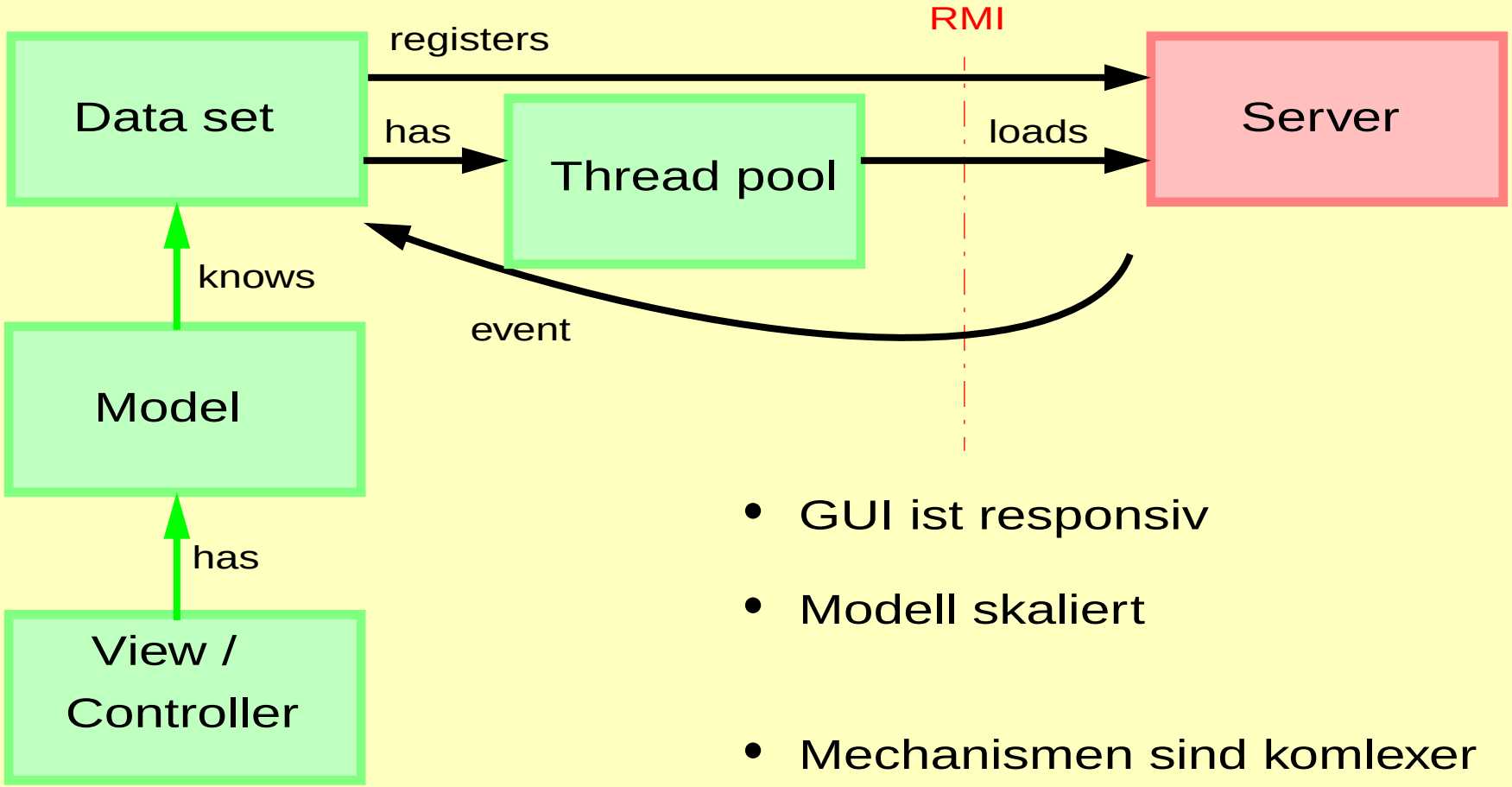
2. RMI encapsulated.



- klarer und unikater Code
- Verantwortlichkeiten getrennt
- Modell skaliert nicht

Graphical User Interface

3. Loading on demand.



- GUI ist responsiv
- Modell skaliert
- Mechanismen sind komlexer

Speicherverwaltung

THE DOCUMENT COMPANY XEROX

Garbage Collection übernimmt Speicherverwaltung !?!?

Aber:
`OutOfMemoryException` ist nicht akzeptabel auf dem Server

=> Speicherverwaltung auf dem Server ist wichtig

Massnahmen, um `OutOfMemoryException` zu vermeiden:

- weniger Objekte im Speicher halten
- Server vor Überlastung schützen

Speicherverwaltung

Weniger Objekte halten

THE DOCUMENT COMPANY XEROX

- Abgearbeitete Jobs und Dokumente aus dem Speicher entfernen
- Jobs “on hold" entfernen
- Weak References auf die Printing Objekte
  und neu aus der Datenbank laden, falls nötig

=> Verbesserte Ausnutzung des vorhandenen Speichers, aber:

`OutOfMemoryException` kommt einfach später

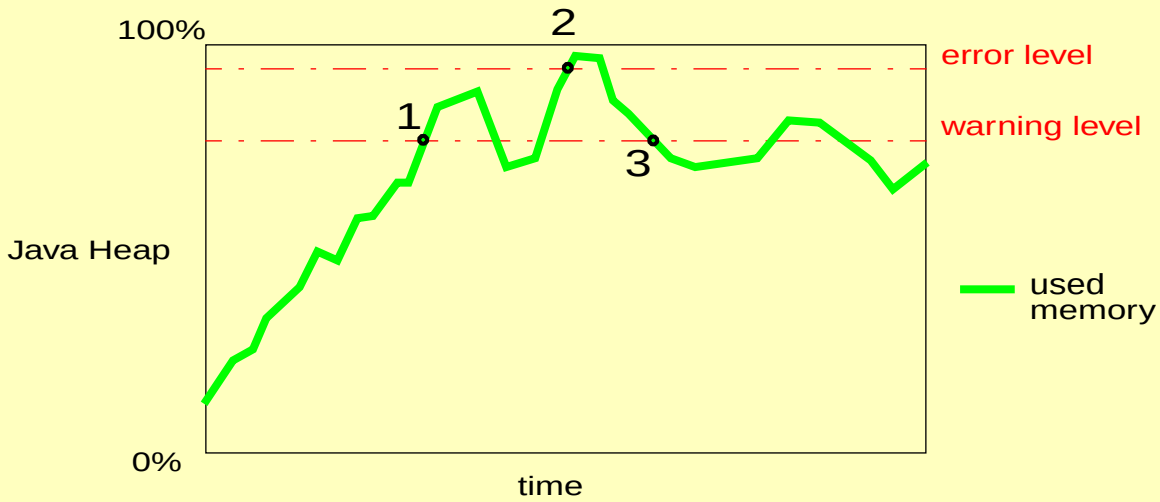
Speicherverwaltung

Aktive Speicherüberwachung

THE DOCUMENT COMPANY XEROX

"MemoryWatcher" als separater Thread

- System.gc() bei Warning Level
- server.disable() bei Error Level



Schlussfolgerungen

THE DOCUMENT COMPANY XEROX

- RMI ist simpel, Distributed Systems nicht
- falls GUI über RMI Daten beschafft: ein mehrstufiges MVC hilft (Kapselung, Komplexität, Performance)
- Speicherverwaltung kriegt man nicht gratis,
  etwas Aufwand ist nötig, trotz garbage collection

Fragen

Demo

THE DOCUMENT COMPANY
XEROX

Besuchen Sie uns an unseren Projekt-Tisch beim anschliessenden Apèro.