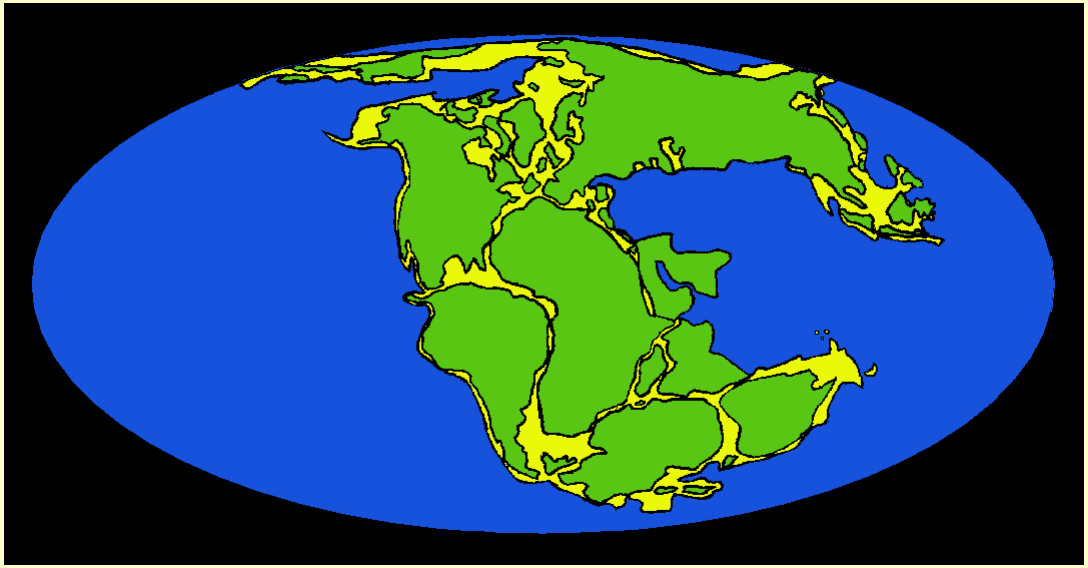
Automatische Verteilung in Pangaea



André Spiegel
Freie Universität Berlin

Einführung

Pangaea ist ein System, das zentralisierte Java-Programme automatisch verteilen kann

- basierend auf statischer Quelltextanalyse
- unter Verwendung beliebiger Middleware
  wie RMI oder CORBA.

→ ein verteilender Übersetzer.

Anwendungsgebiete:

- interaktive, client/server-artige Web-Applets
- parallele Programmierung.

Gliederung

- Motivation: Warum automatische Verteilung?
- Pangaea im Überblick
- Technische Einzelheiten
  - Objektgraph-Analyse
  - Codegenerierung für eine Middlewareschicht
- Eine Fallstudie

Motivation

Warum automatische Verteilung?

- erleichtere die Erstellung verteilter Programme
- entkoppele Verteilung und Anwendungslogik

→ Verteilungstransparenz konsequent

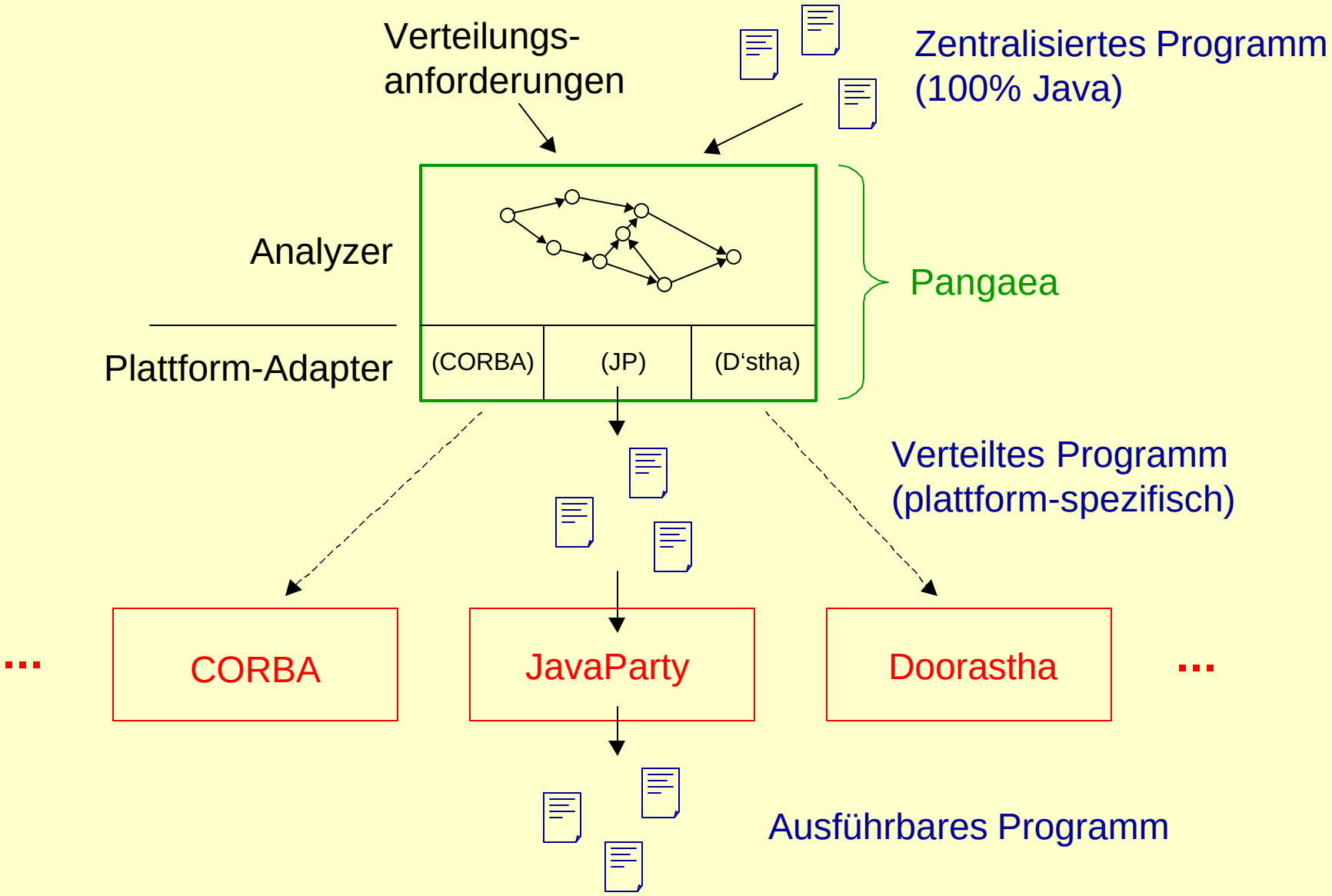
Warum statische Analyse?

- findet Optimierungen, die für eine rein laufzeit-basierte Automatik unmöglich sind
- Beispiel: konstante Objekte (immutable objects)

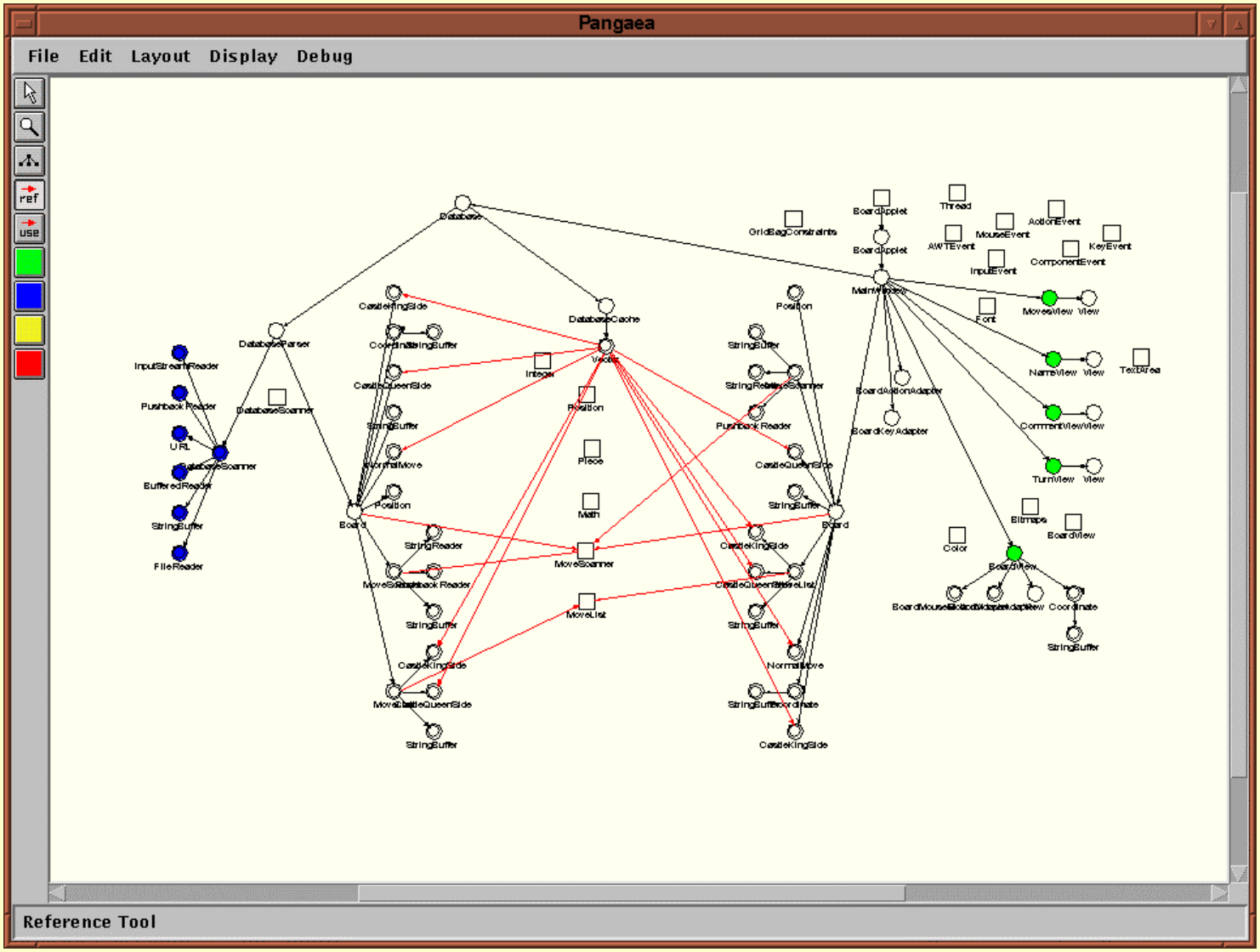
Ziele

- zunächst fast so gut sein wie der Programmierer...
- ...später besser (vgl. Compilertechnik)

Architektur



Die Benutzerschnittstelle



Objektgraph-Analyse

Das Ziel: Gewinne durch statische Analyse einen Graphen

- der Objekte, die zur Laufzeit existieren werden
- der Benutzungsbeziehungen zwischen diesen Objekten (wer ruft wen?)

Existierende Ansätze: Typinferenz, Points-to Analyse, ...

- betrachten Typen, nicht Objekte
- ein statisches Programm manipuliert passive Daten

Unser Ansatz: eine objekt-orientierte Sicht des Problems

Eine Beispielanalyse

```
public class Main ...

public class Producer ...

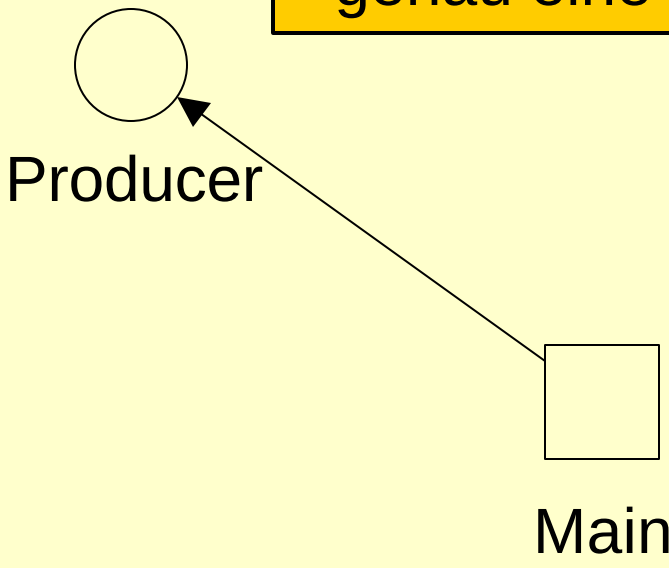
public class Data ...

public class Buffer ...

public class Consumer ...
```

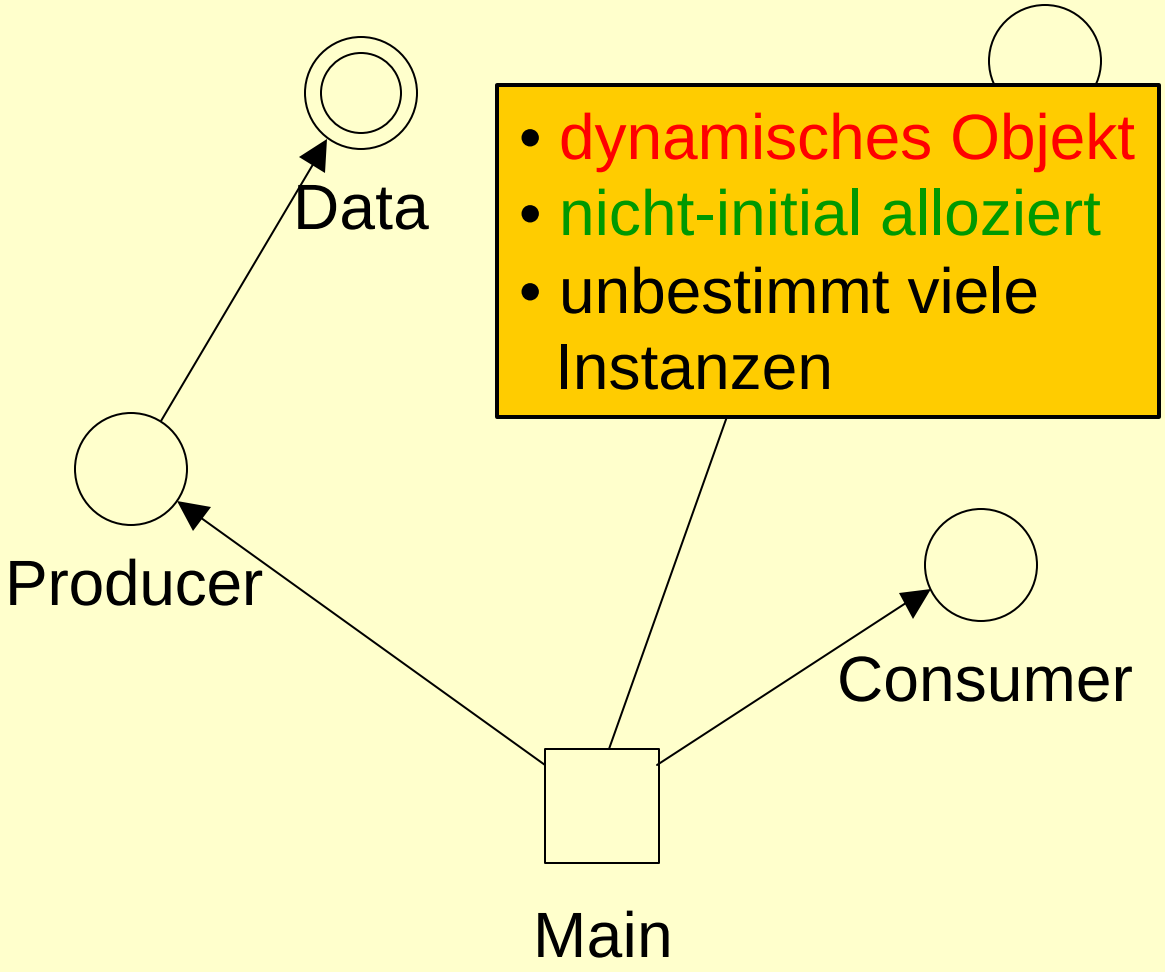
Eine Beispielanalyse

- dynamisches Objekt
- initial von `Main` alloziert
- genau eine Instanz

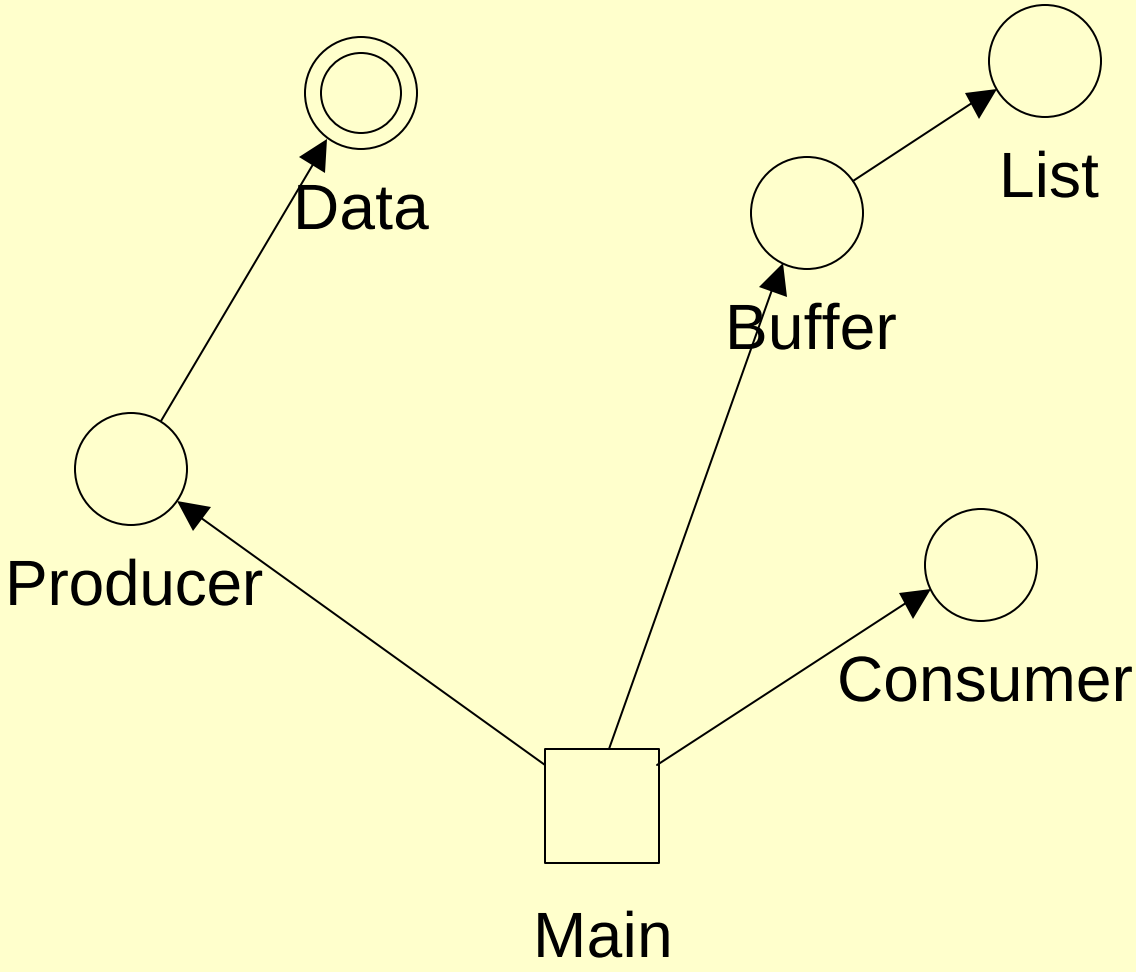


- statisches Objekt
- implizit alloziert
- genau eine Instanz

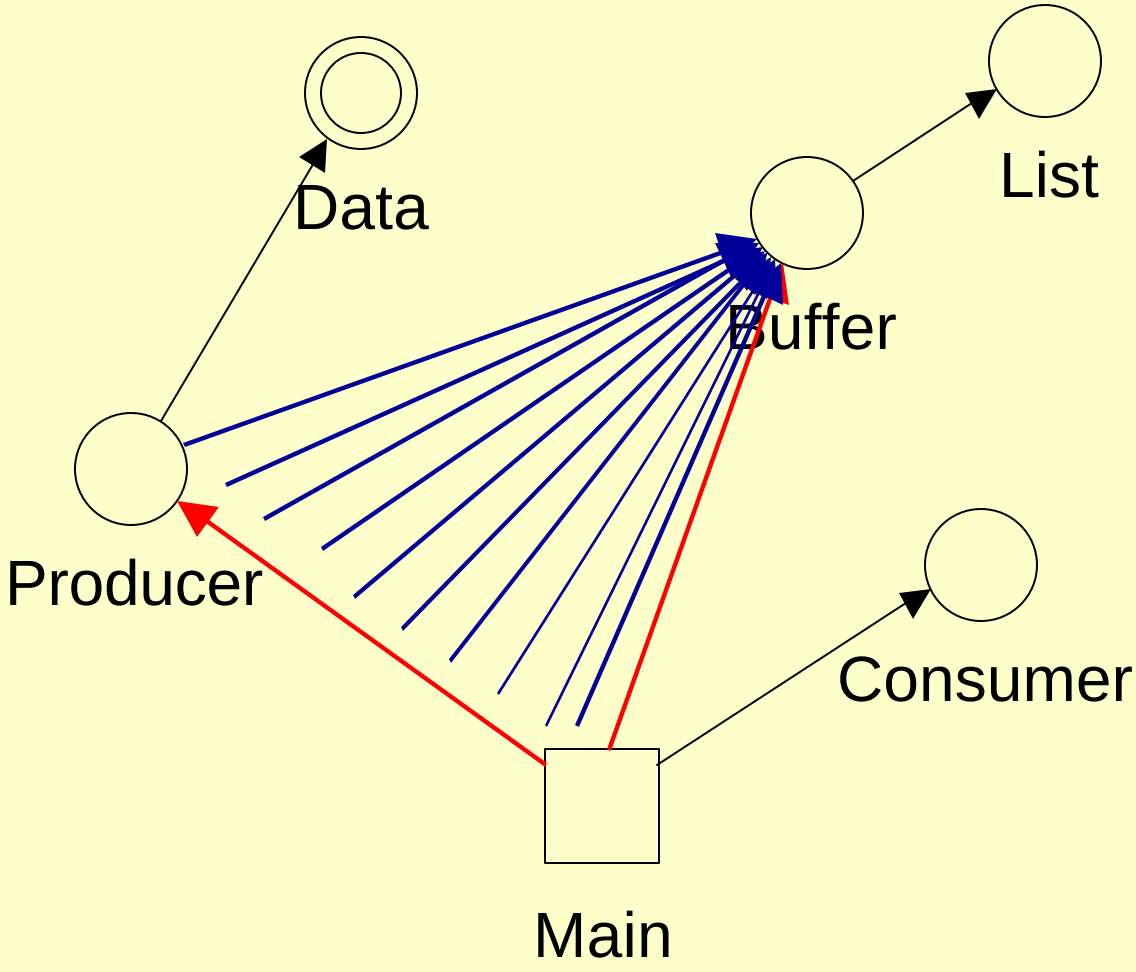
Eine Beispielanalyse



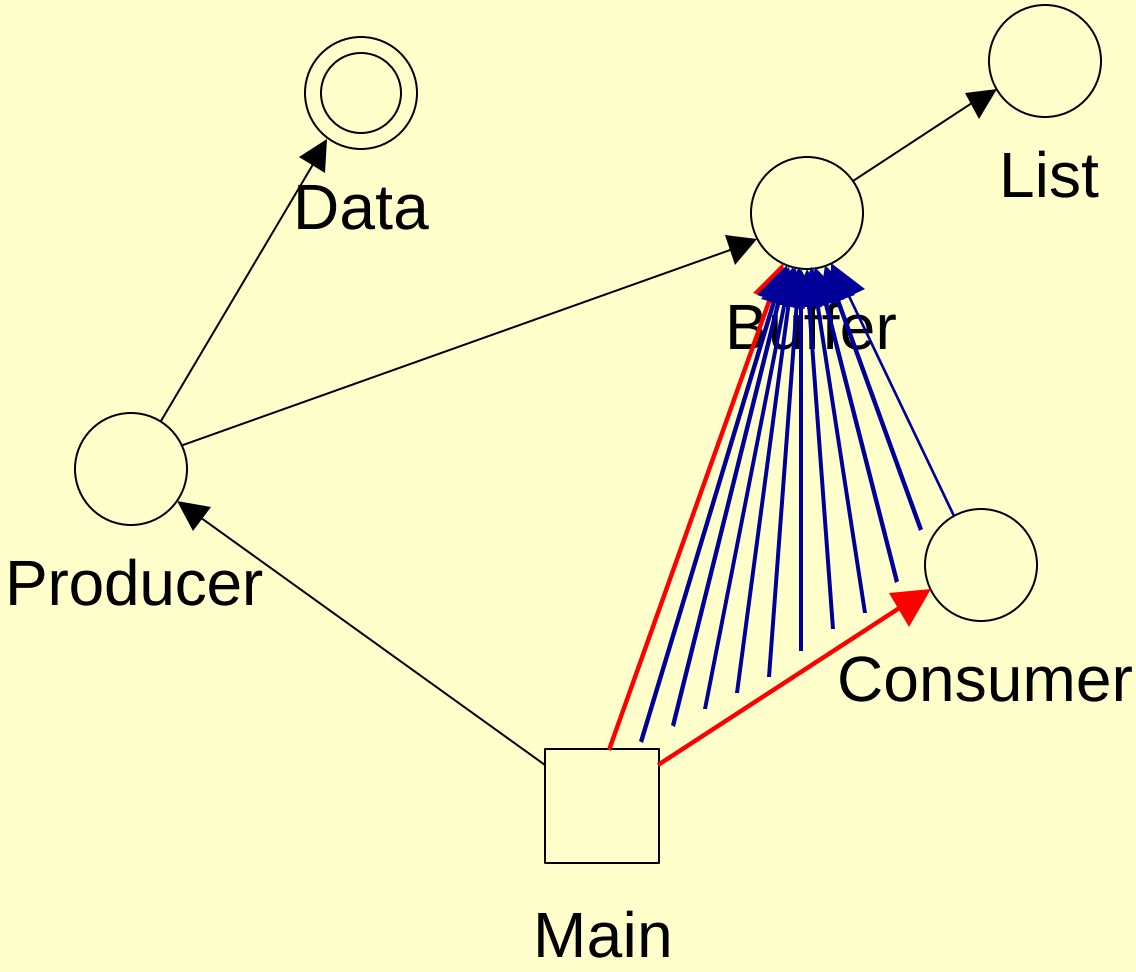
Eine Beispielanalyse



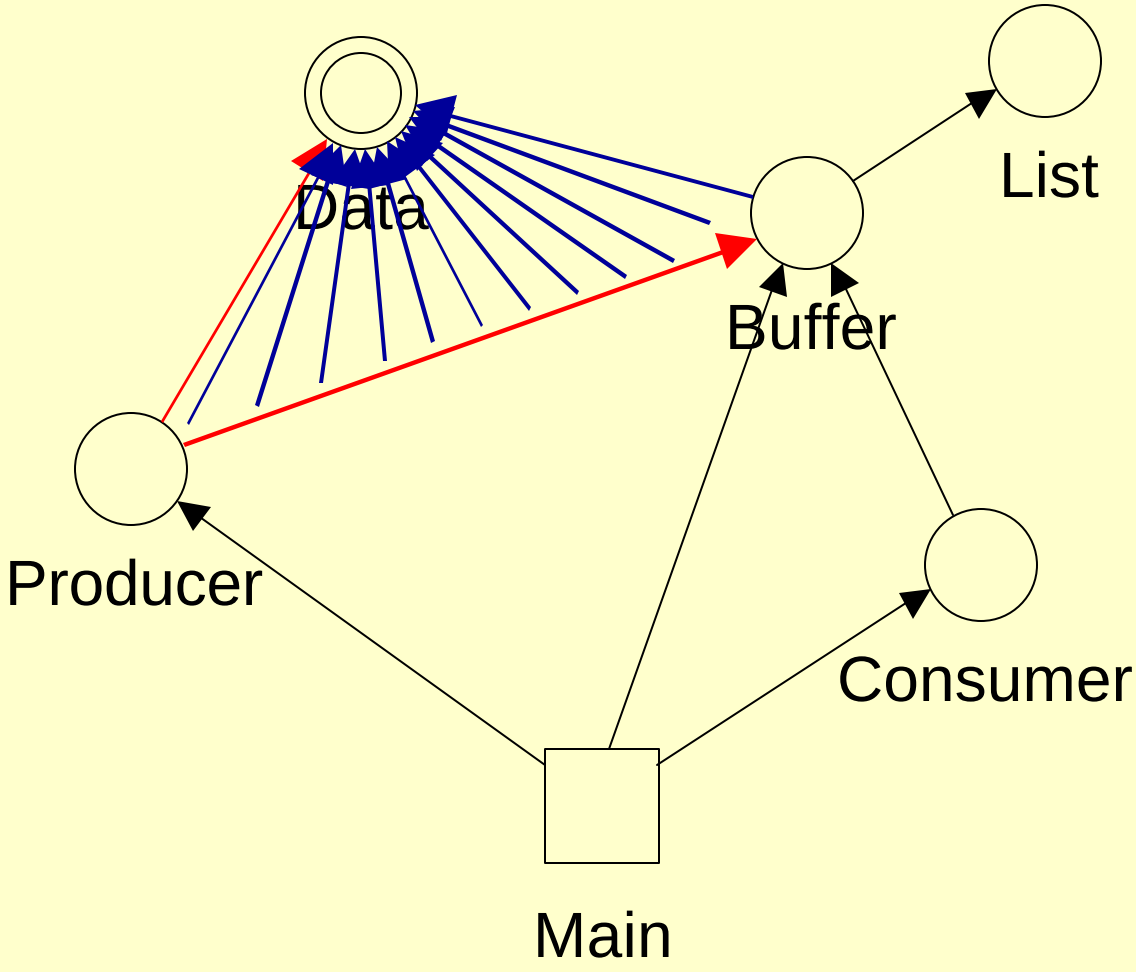
Eine Beispielanalyse



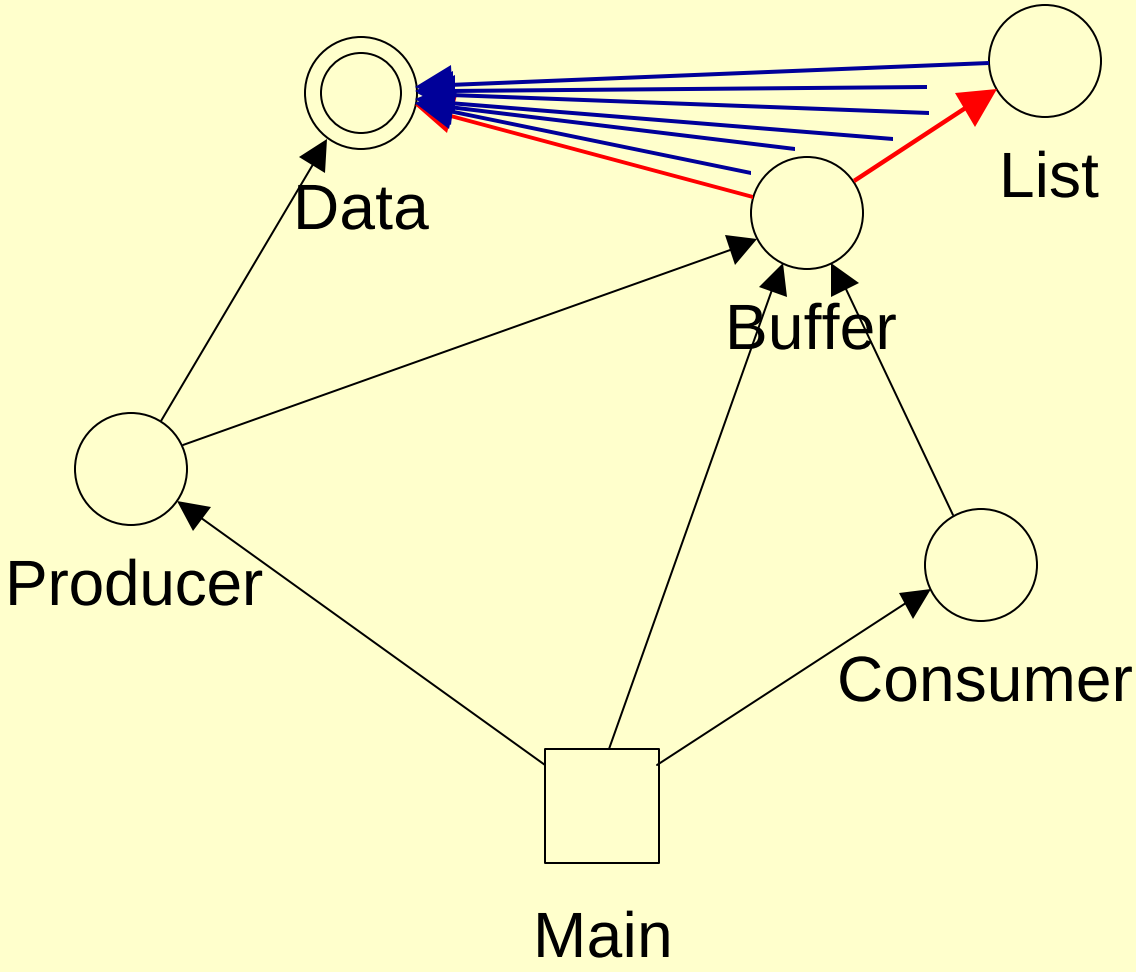
Eine Beispielanalyse



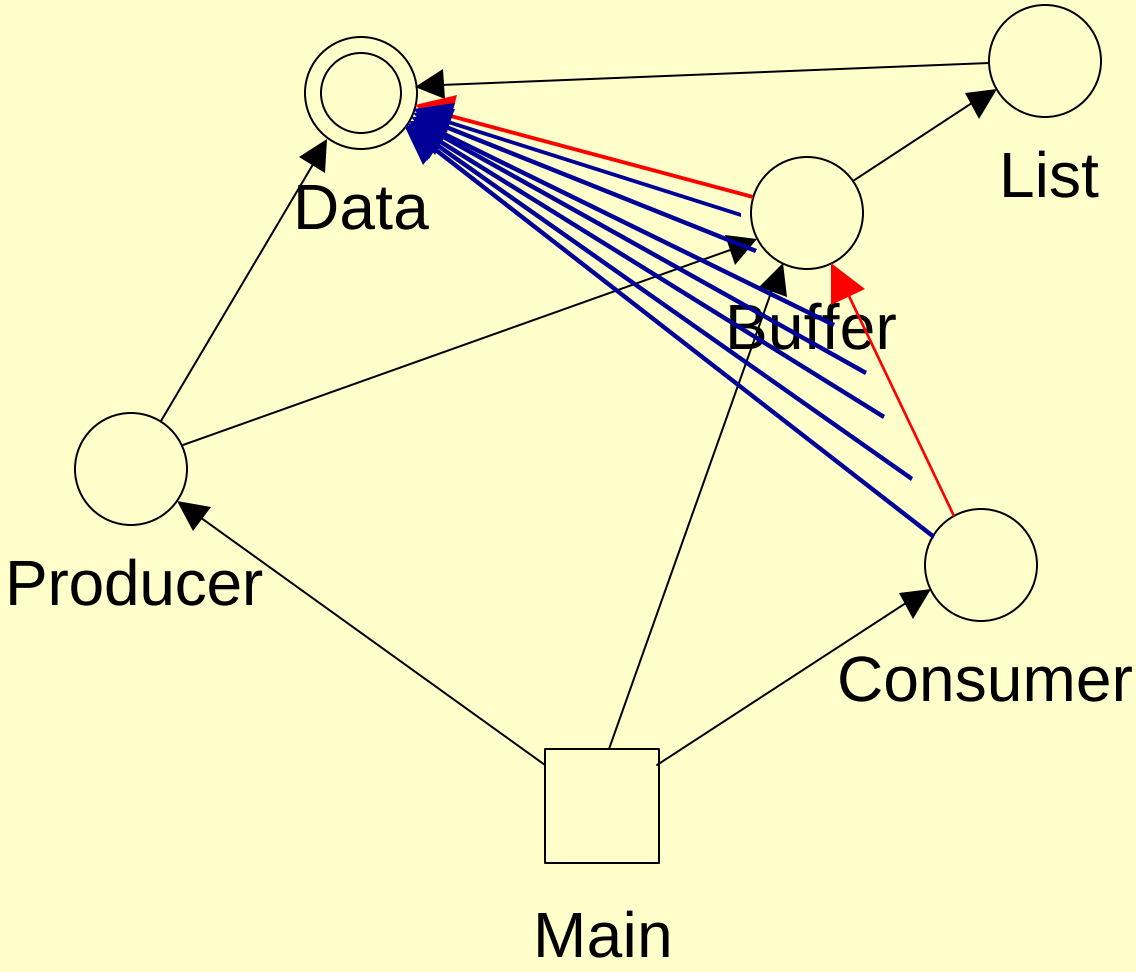
Eine Beispielanalyse



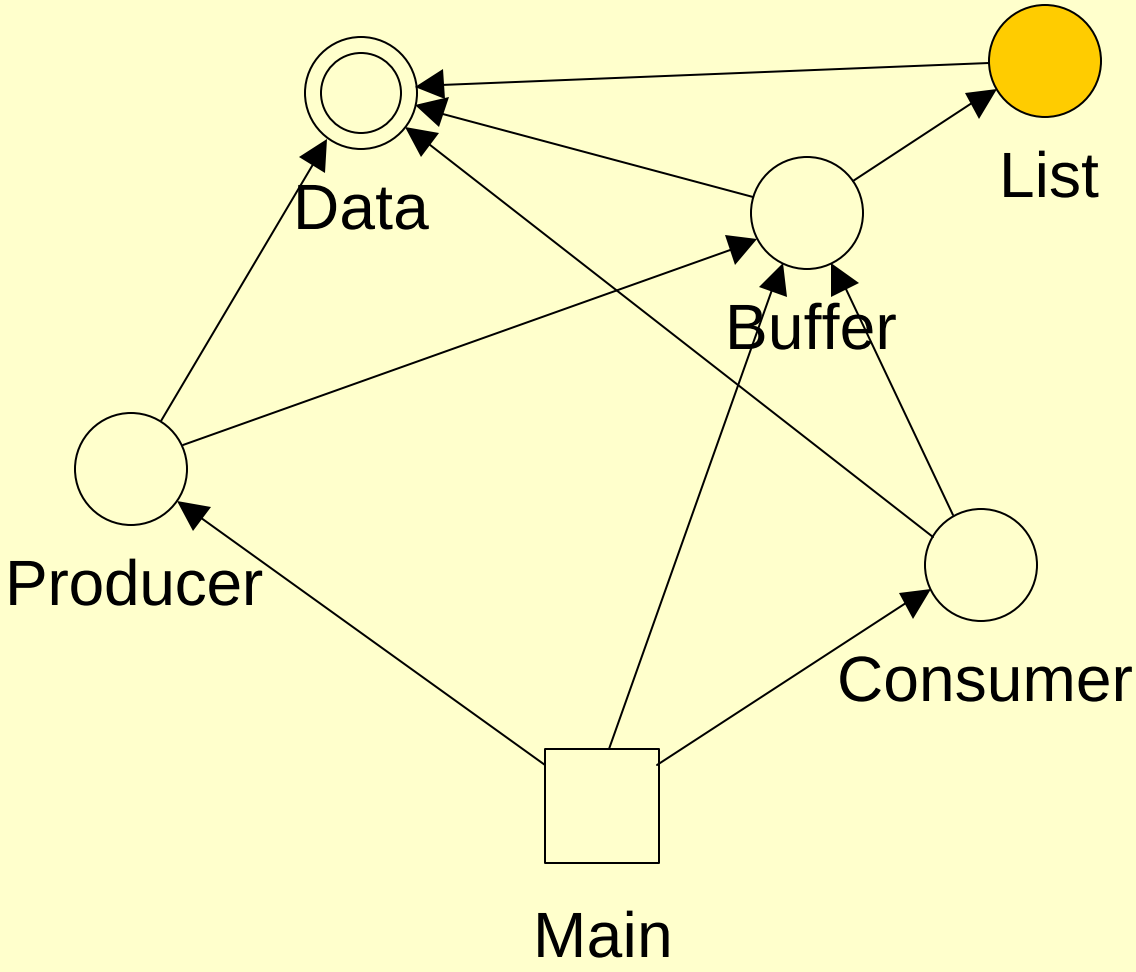
Eine Beispielanalyse



Eine Beispielanalyse



Eine Beispielanalyse



Eigenschaften des Algorithmus‘

- Für reale Programme: polynomielle Komplexität
- Implementiert und getestet für Programme mit bis zu 10.000 Zeilen (Rechenzeiten im Minutenbereich)
- Detailauflösung hinreichend zur Verteilungsanalyse

Codegenerierung

Das Ziel:

- realisiere die Verteilungsstrategie, die der Analyzer verlangt, auf der gegebenen Middleware
- unter Wahrung der zentralisierten Semantik.

Einfachster Fall:

- mache einige Klassen fernaufrufbar („Globalisierung“)
- ändere einige new-Anweisungen in Fernerzeugungen.

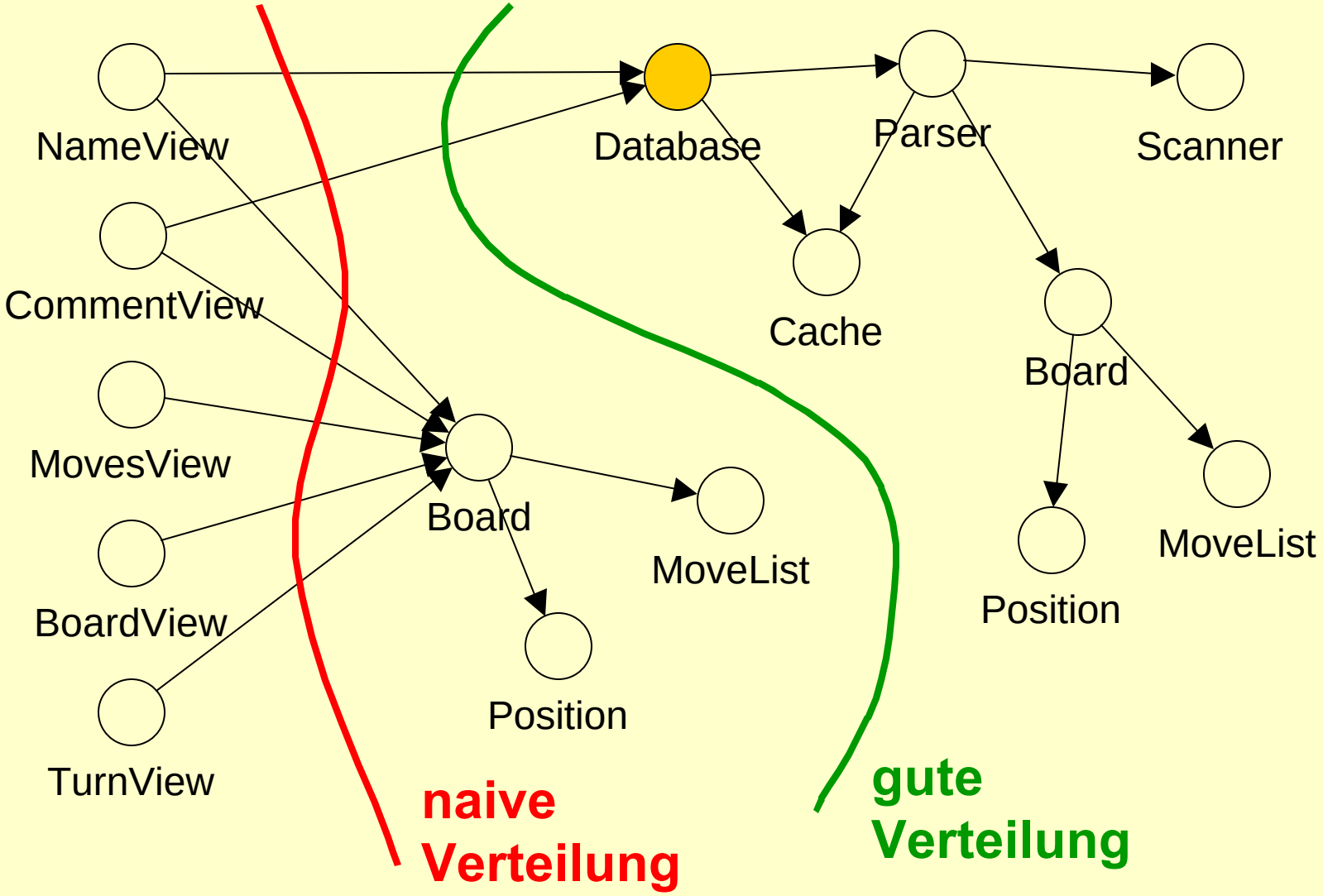
Semantikerhaltung:

- durch Codetransformation

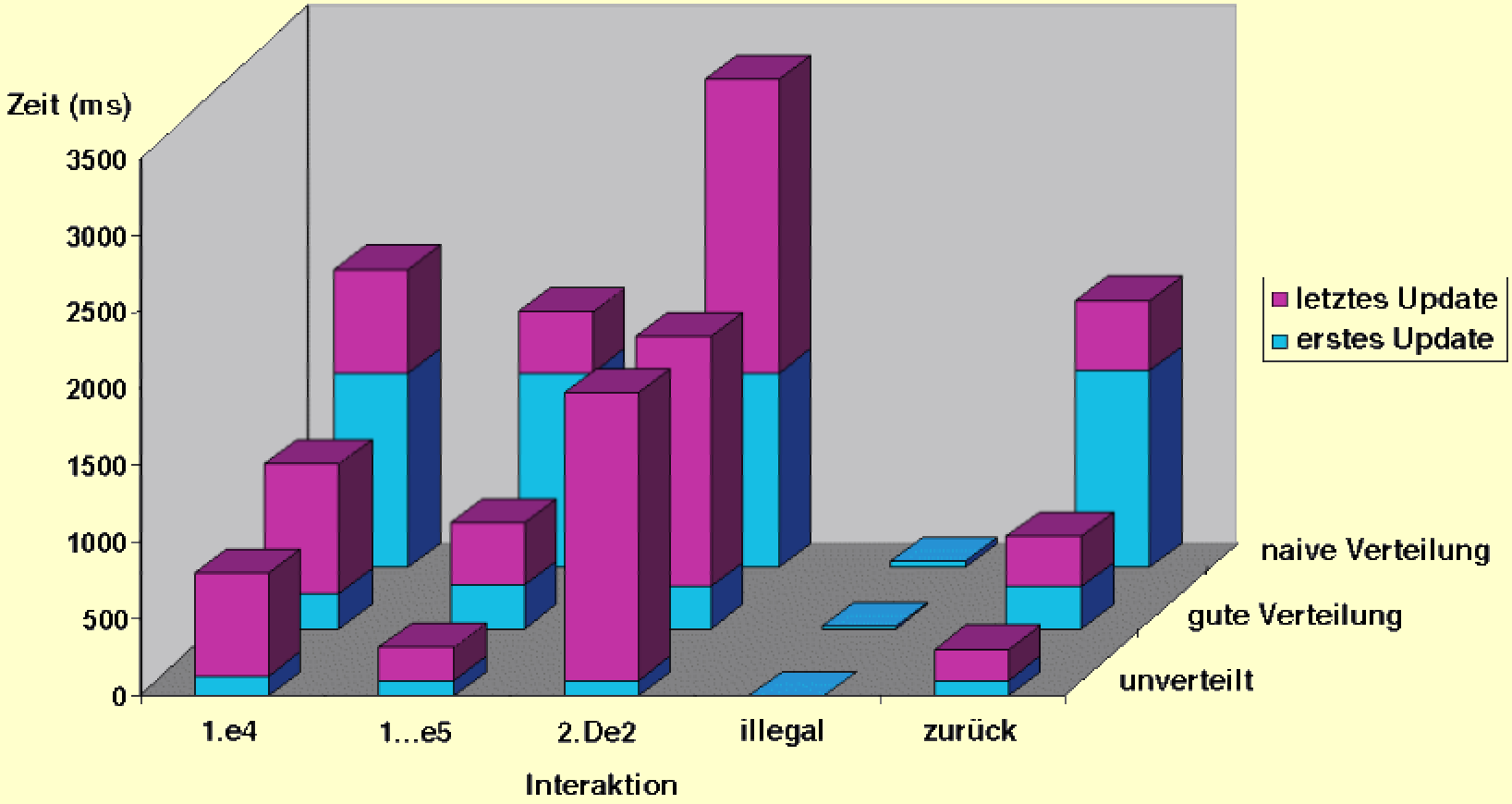
Eine Fallstudie



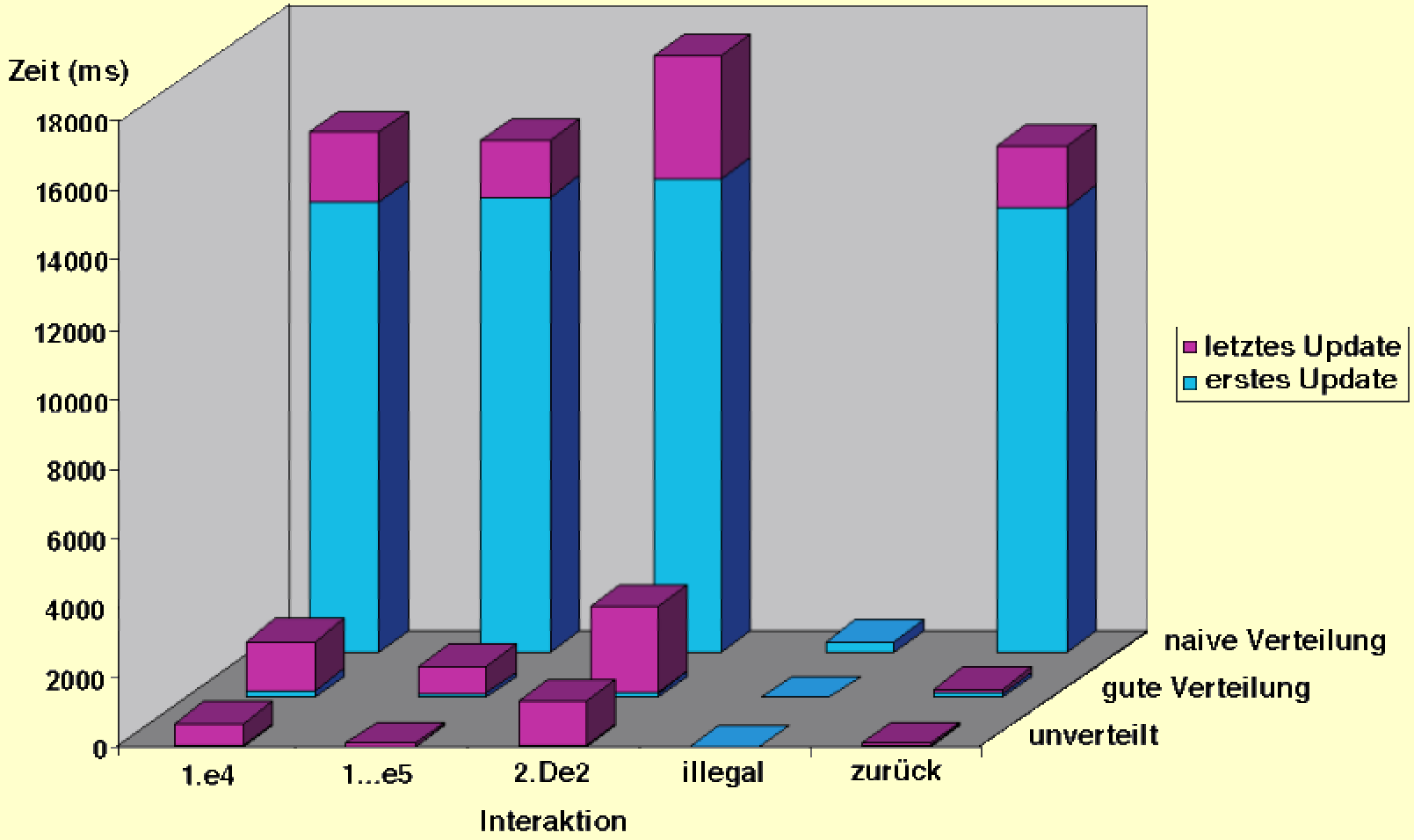
Verteilungsanalyse



Verteilung via Ethernet (10 MB/s)



Verteilung via Modem (28.8 kB/s)



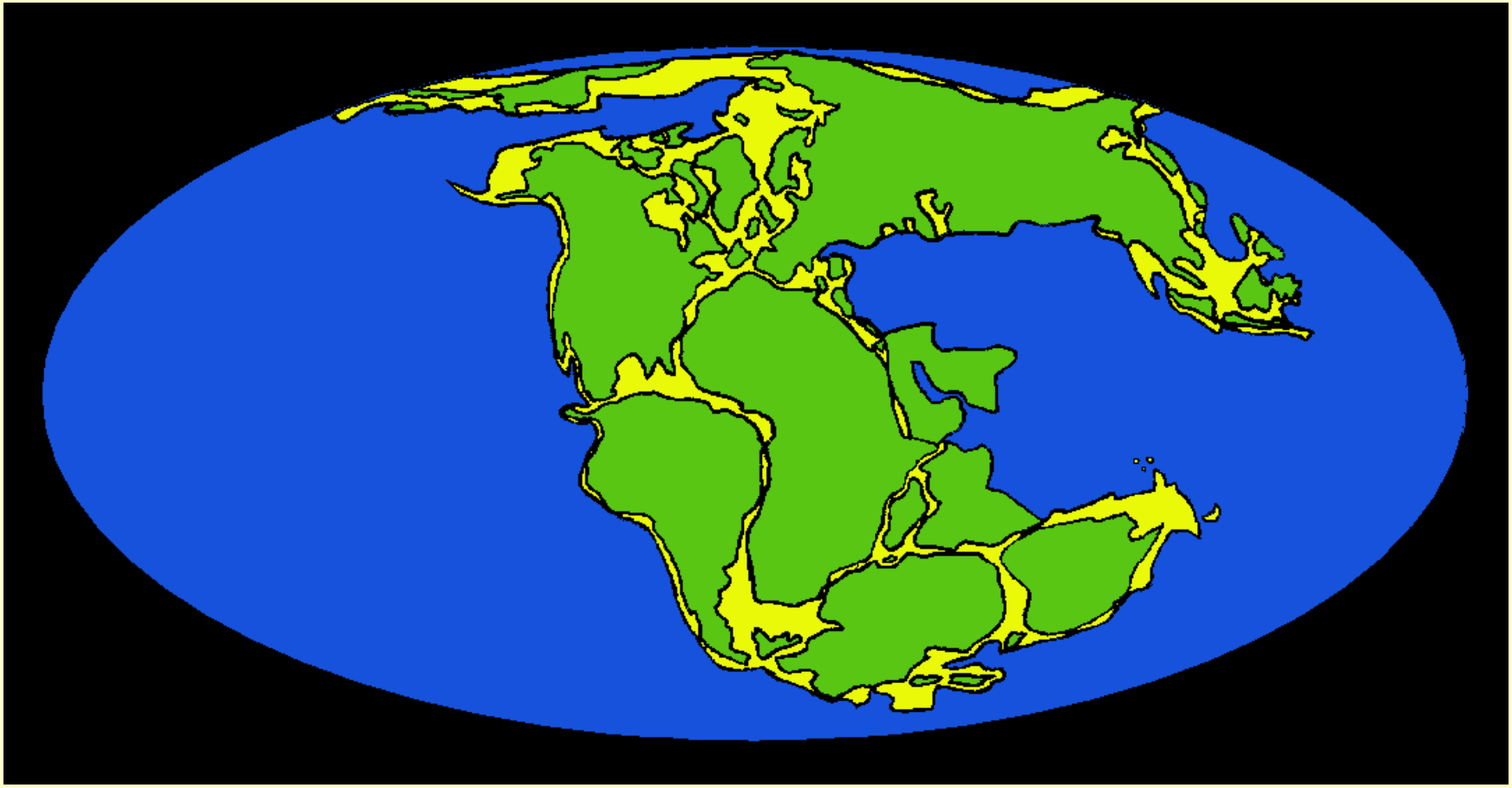
Zusammenfassung

- Pangaea ist ein verteilender Übersetzer: Java-Programme werden für verteilte Maschinen übersetzt, die durch beliebige Middleware realisiert sein können.
- Die Arbeit des Programmierers wird erleichtert; eine zukünftige Automatik wird dem Programmierer vielleicht sogar überlegen sein.
- Statische Analyse erlaubt es, bessere Verteilungs-entscheidungen zu treffen, als es für eine rein laufzeit-basierte Automatik möglich wäre.

Projektstatus



- Implementiert:
  - Objektgraph-Analyse
  - Graphische Oberfläche
  - Rudimentäre Codegenerierung (JavaParty)
- Geplante Plattform-Unterstützung:
  - JavaParty, CORBA, Doorastha, Jackal, ...
- Alpha-Release ist frei verfügbar (GPL)



www.inf.fu-berlin.de/~spiegel/pangaea