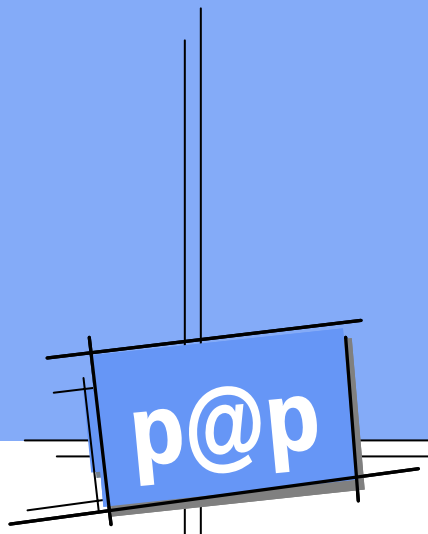
Consistency Management for Peer-to-Peer based MMVEs



Gregor Schiele[1], Richard Süselbeck[1], Arno Wacker[2], Tonio Triebel[1], and Christian Becker[1]

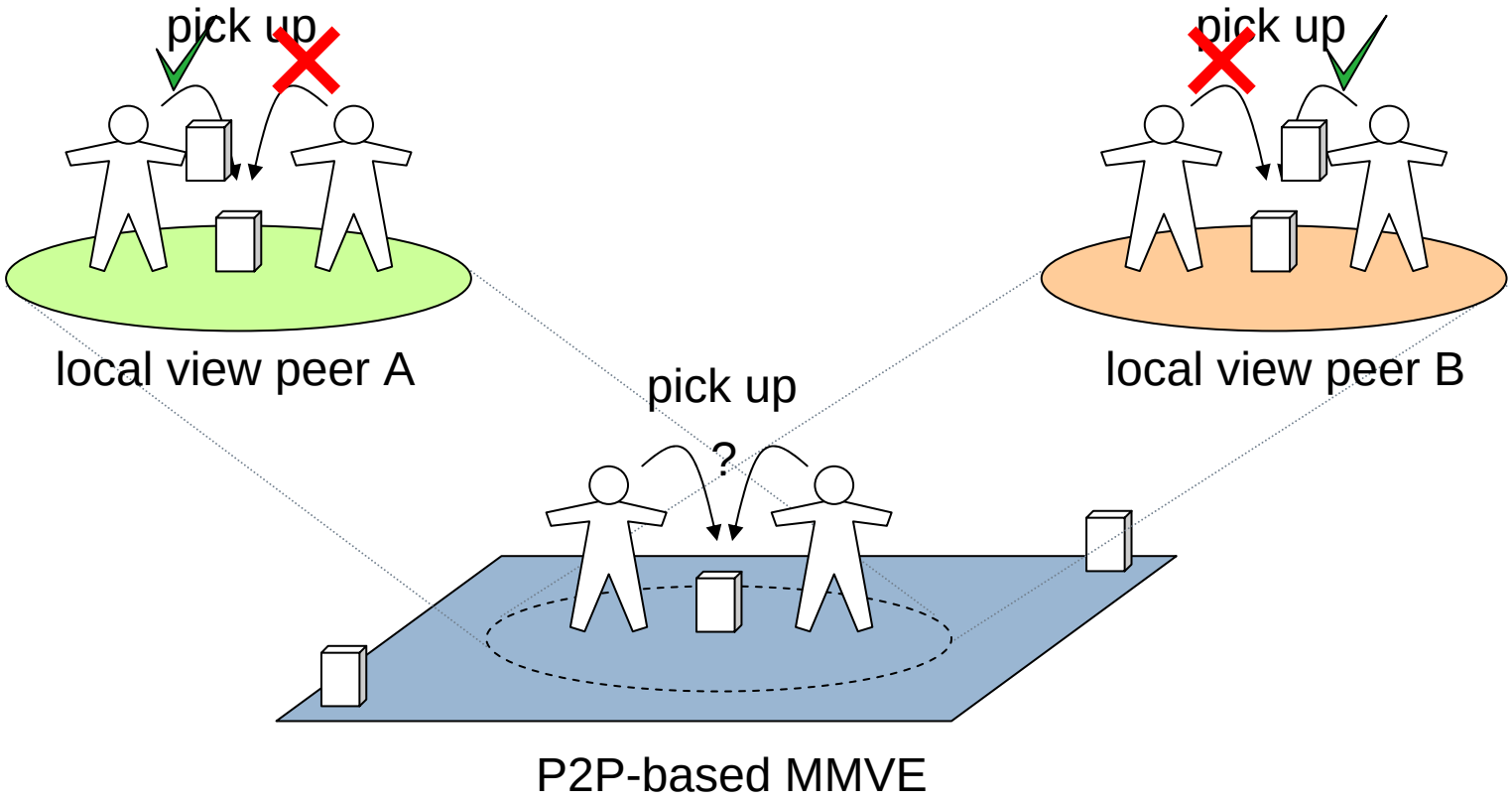
1) University of Mannheim
Mannheim, Germany

2) University of Duisburg-Essen
Duisburg, Germany

Overview

- Scenario & goal
- The peers@play project
- Our approach
- System architecture
- Conclusion & future work

Scenario


pick up
pick up
local view peer A
local view peer B
pick up
?
P2P-based MMVE

Motivation

- Consistency management required
  - High responsiveness
  - Seamless operation
  - Resource efficiency
  - Distributed control

→ high complexity for developers

How can we support developers with this?

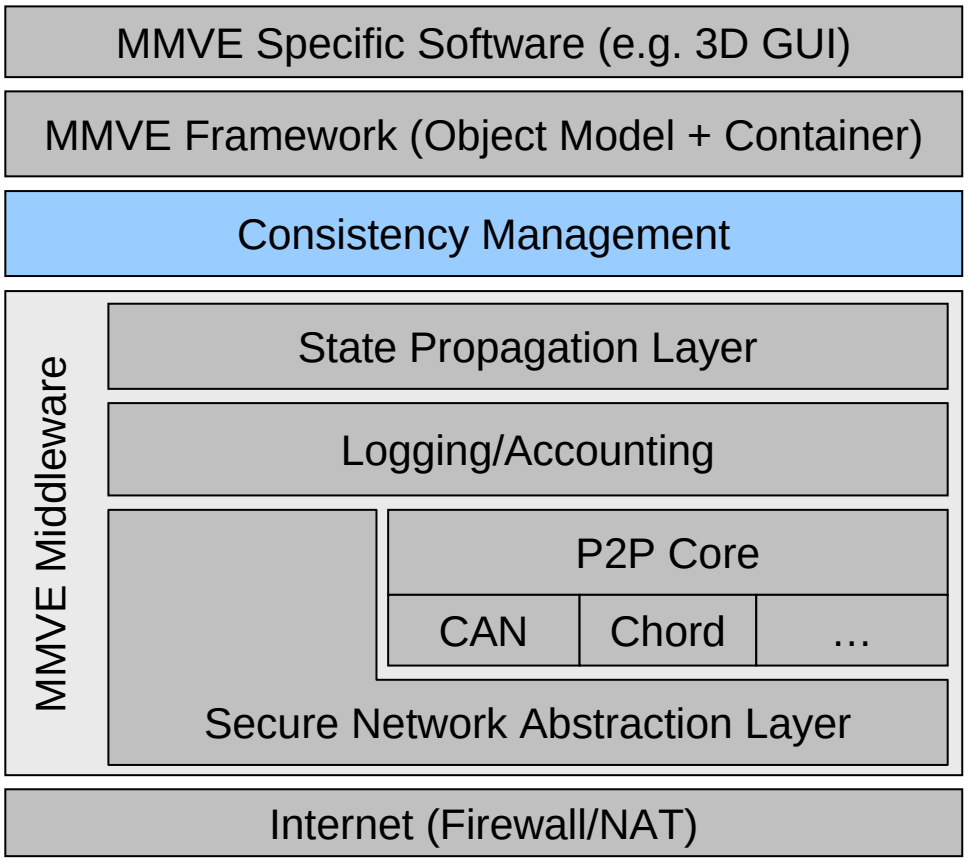
Our Goal

Support MMVE developers by providing a generic consistency management infrastructure that is…

- **Flexible**
  - Support different behaviors w.r.t. consistency for different situations and activities
- **Adaptive**
  - React to fluctuating resource availability & system load
- **Extensible**
  - Allow developers to add new behaviors

The Peers@Play Project

- **Goal**: conceptual framework for p2p MMVEs
  - algorithms
  - protocols
- **Main topics**:
  - scalability
  - consistency
  - security
- **Conducted by**:
  - Univ. Mannheim
  - Univ. Duisburg-Essen
  - Univ. Hannover



Our Approach

Provide generic consistency mngt infrastructure

Main concepts:

- Selectable synchronization protocol
- Reflection API
- Consistency plugins
- Consistency sessions
- Dynamic relocation

Selectable Synchronization Protocols

- Infrastructure offers different synchronization protocols (providing different consistency models):
  - Laissez faire
  - Conservative (strict synchronization)
  - Optimistic (rollback)
- Application specifies required consistency
  - Tradeoff between delay / overhead and inconsistencies
  - Depends on: type of update, update context, system load
  - E.g.: moving vs. picking up, passing by vs. fighting
- Infrastructure selects suitable protocol

Reflection API

- Choice of consistency may depend on current system work load & resources:
  - Willing to sacrifice consistency for performance?
  - Low load → little delay → high consistency
  - High load → large delay → low consistency
- Reflection API to query current work load
  - Expected delay for given consistency
  - Predicted using historical data (log)
- Log exchange between peers
  - Better prediction for moving peers

Consistency Plugins

- Encapsulate synchronization protocols
- Different MMVEs may require different consistency models & synchronization protocols
  - Highly optimized for specific kinds of updates, e.g. involving electronic payments
  - New & better protocols
- Developer can extend infrastructure with plugins
  - Set of plugins selected at development time

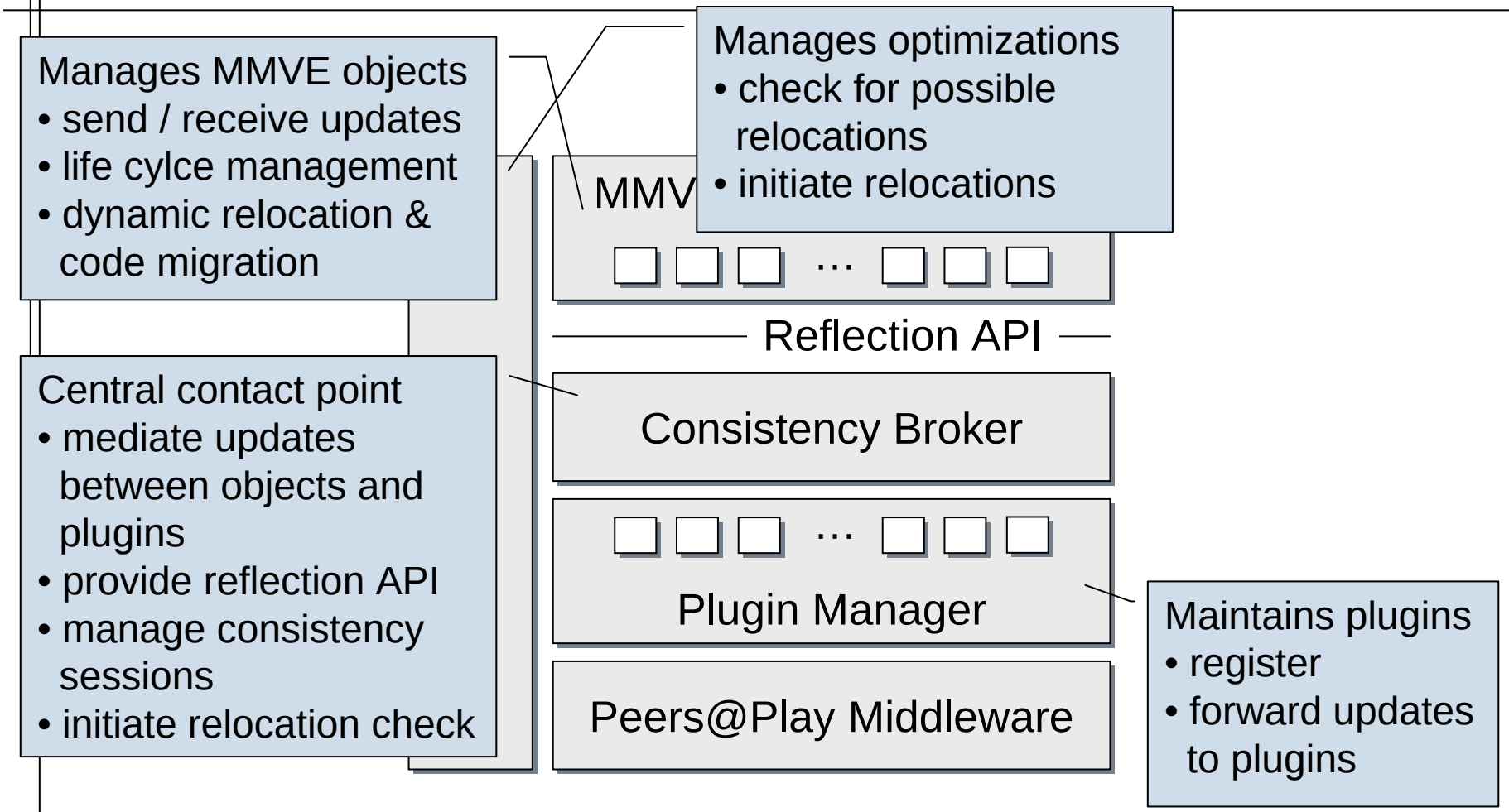
Consistency Sessions

- Groups updates into common context
- Optimistic synchronization: rollback possible
  - E.g.: object could not be taken → undo using it
- Disrupts seamless MMVE operation
  - May confuse and annoy users
- Perform partial rollback using consistency session
  - Allows application to specify dependencies between updates
- Also: allows choosing multiple consistency models for update
  - Users fighting each other: create session & select high consistency for movement updates
  - All others: low consistency

Dynamic Relocation

- Synchronization delay depends on placement of update sources in P2P network
  - User in USA interacting with NPC executed on peer in Asia
- Location of update sources modifiable
  - Peer executing NPC can be changed
- Dynamic relocation of MMVE objects (e.g. NPCs)
  - Due to high overhead: triggered by application
  - Determined and executed by infrastructure

System Architecture



Conclusion & Future Work

- Consistency crucial for MMVEs
  - Tradeoff: consistency vs. delay & overhead
- Consistency infrastructure designed
  - Basic support for developers provided
  - Implementation underway
- Future Work:
  - Detailed evaluation
  - Support for inter-plugin dependencies
  - Extended support for automated optimization decisions

Thank you for your attention!

- **Contact data:**

  Gregor Schiele, University of Mannheim,
  Email: gregor.schiele@uni-mannheim.de

- **Project webpage:**

  http://www.peers-at-play.org