

SLiM – A Lightweight Environment for Synchronous Collaborative Modeling

Christian Thum, Michael Schwind, Martin Schader



UNIVERSITY OF
MANNHEIM



Agenda

Motivation and Research Objectives

Design Aspects

The SLiM Collaboration Environment

Conclusion



Motivation:

Challenges in Collaborative Software Engineering

Software design is an inherently collaborative activity

- **Innovative tools** are needed for tackling the challenges of distributed development:
 - Rising number of **outsourcing** and **offshoring projects**
 - Increased **dynamics** in the economic environment and evolving organizational structures
 - **Informal collaboration** in open source projects
 - Distributed teams with **heterogeneous system environments**



Motivation:

Challenges in Collaborative Software Engineering

“

In particular, **support for highly synchronous activities** like software design has to be significantly enhanced. [...] A shared **whiteboard for collaborative elaboration of UML models** has to address these challenges to make virtual collaboration around design artifacts truly effective.”

[Sengupta et al. 2006, A Research Agenda for Distributed Software Development]

“

[There is a] need for **tight integration** of **new and existing web-based and desktop-based** environments [and] **broader participation** in the entire development process.”

[Whitehead 2007, Collaboration in Software Engineering: A Roadmap]



Research Objectives

Tool support for the development of a common understanding of a system's design in agile, distributed scenarios

- The first stage of our research mainly focuses on the **technical feasibility of a lightweight tool** that runs in unmodified browsers:
 - Lower the technical entry barriers for participating in the modeling process
 - Avoid technical issues impeding clients to be integrated into the collaborative environment
 - Facilitate ad hoc collaboration

Design Aspects

- Traditional approaches to interconnecting developers require the installation, configuration, and maintenance of the same software on all client systems, which is time-consuming and costly.



Ad hoc availability and straightforward accessibility

- Comprehensive web-based development platforms span the entire development process and form the central-infrastructure for coordinating and documenting project activities.



Integration into existing web-based platform

Design Aspects

- In many cases, multiple modeling tools have to be used because not all modeling needs are met by a single tool.



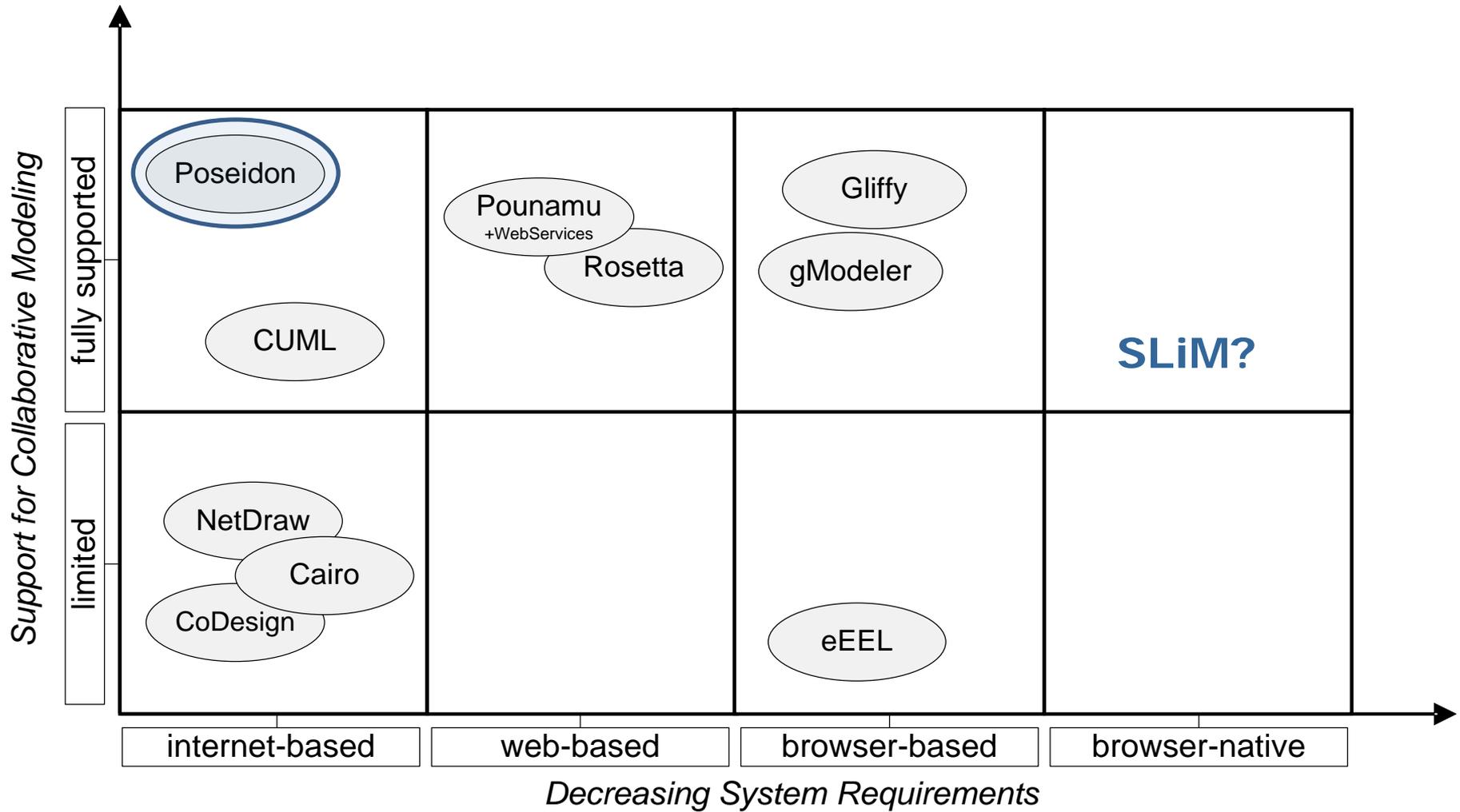
Interoperability with desktop-based modeling tools

- Central non-functional requirements crucial for the aptitude of a collaborative UML modeling tool:



Flexible access, ad hoc availability, interoperability, integration into existing web-based platforms.

Related Work



SLiM – Synchronous Lightweight Modeling

Zero-install and on demand availability: web access and a modern browser* are the only requirements

XMI - Export/Import to desktop based modeling tools like

Changes to the diagram are visible in real-time.

Allows multiple users to simultaneously work on the same diagram in a shared environment

The screenshot shows the 'Browser Native Collaborative Modeling Tool' running in Mozilla Firefox. The interface includes a menu bar (Datei, Bearbeiten, Ansicht, Chronik, Lesezeichen, Extras, Hilfe), a browser address bar, and a toolbar. The main workspace displays a UML class diagram with classes: Person (base class), Student (subclass of Person), Thesis, and Chapter. Student has attributes 'matrikelNr' and methods 'getMatrikelNr()' and 'setMatrikelNr()'. Thesis has a note: 'Writing a thesis is a time consuming and challenging endeavor.' Chapter is associated with Thesis via a 'consists of' relationship. A 'University' class is also present. A right-hand panel shows the 'Student' class details, including its name, stereotype, attributes (matrikelNr), and operations (getMatrikelNr, setMatrikelNr). A chat log at the bottom shows a conversation between Lisa and Bart regarding session state and modeling.

Browser Native Collaborative Modeling Tool

Collaborate Online Work Offline

Person

- name
- + getName()
- + setName()

Student

- matrikelNr
- + getMatrikelNr()
- + setMatrikelNr()

Thesis

Chapter

University

Attributes

Modifier	Name	Type	Default
private	matrikelNr		

Operations

Modifier	Name	Type	Abstra
public	getMatrikelNr		
public	setMatrikelNr		

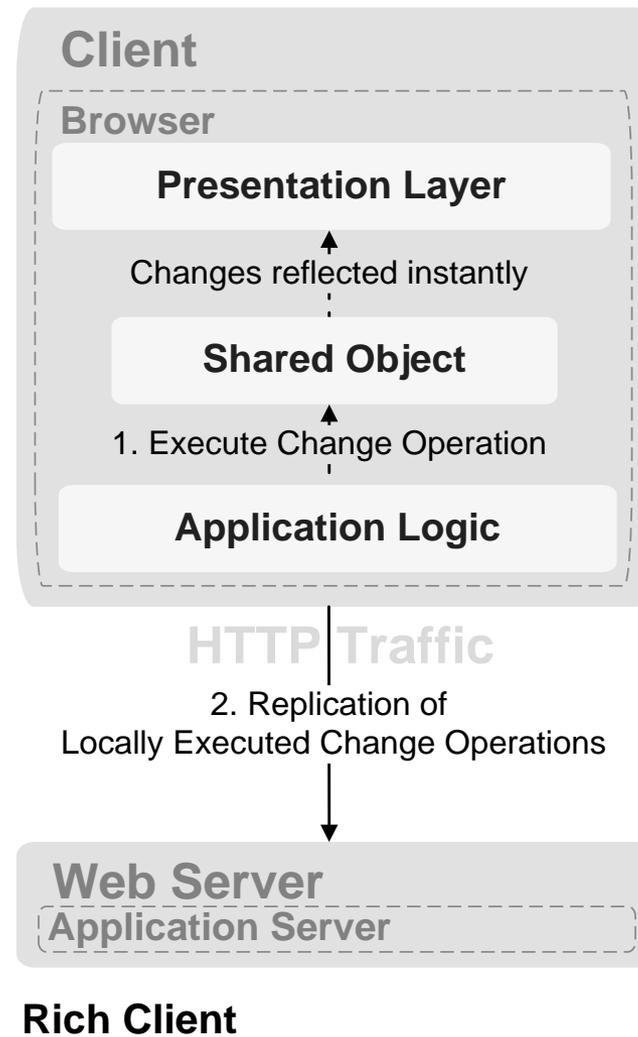
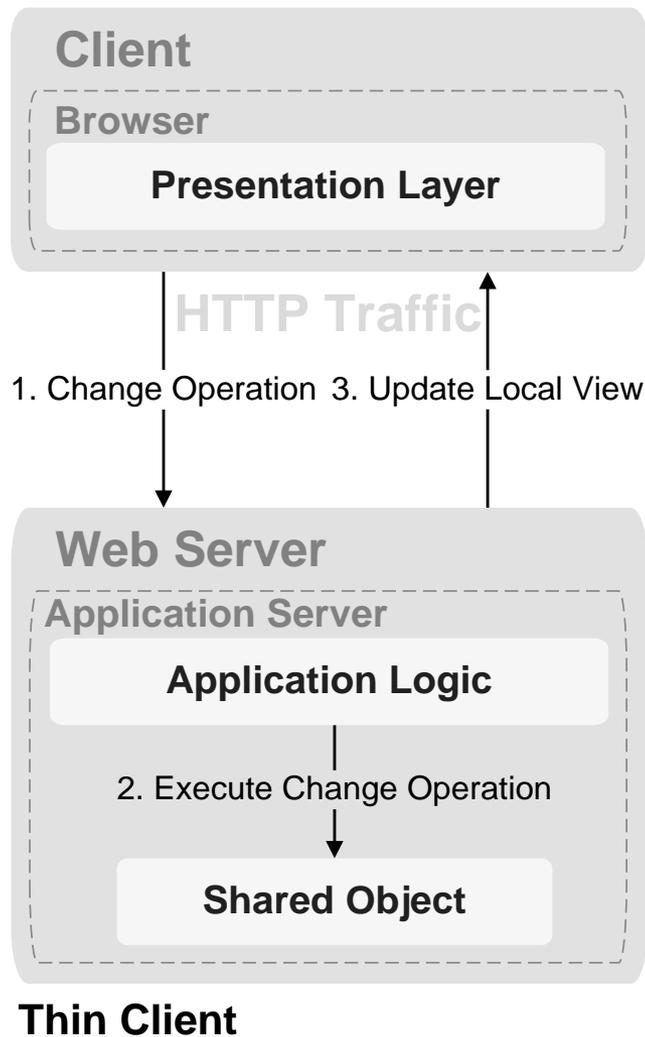
Chat:

- Lisa Bart: Bart has joined
- Bart Bart: requests current session state.
- Lisa: received request from server to serialize local state.
- ... publishes local state.
- Bart: received transmitted state from Server to apply locally.
- Lisa: Hello Bart, could you help me to model this sophisticated diagram?
- Bart: Sure Lisa, using this lightweight collaborative modeling tool anything is possible ;-)





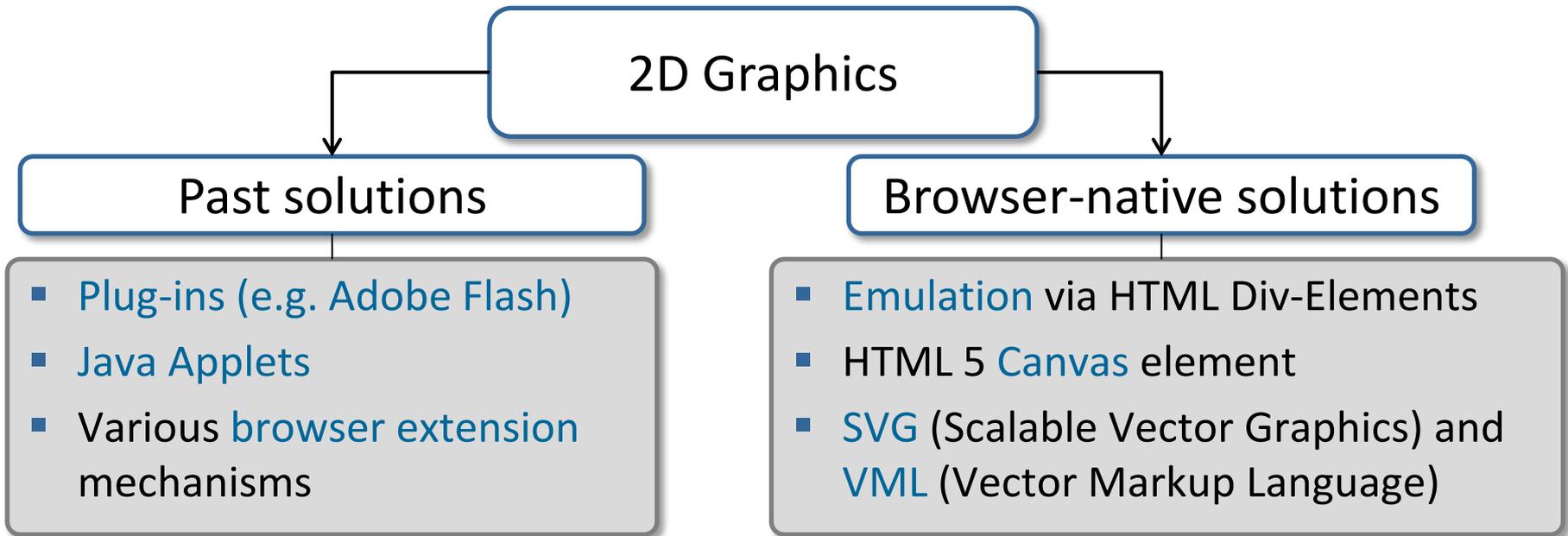
The SLiM Collaboration Environment: Architecture Alternatives



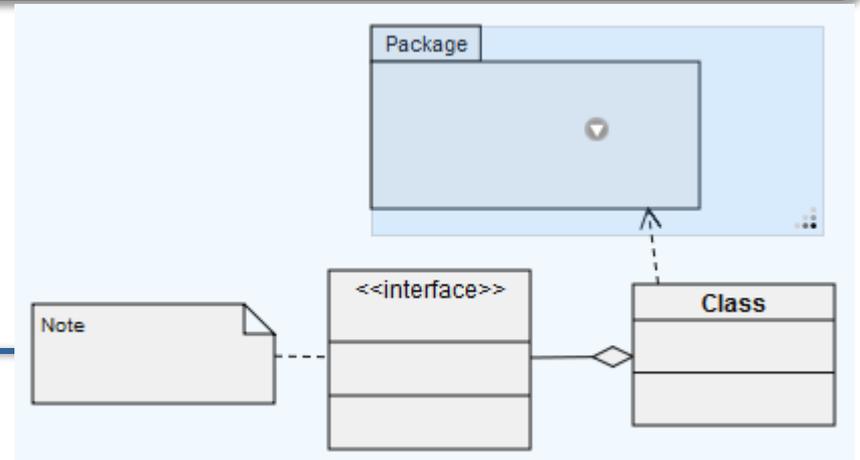


The SLiM Collaboration Environment: Diagram Editor

Challenge: Visualization of interactive UML elements in the browser



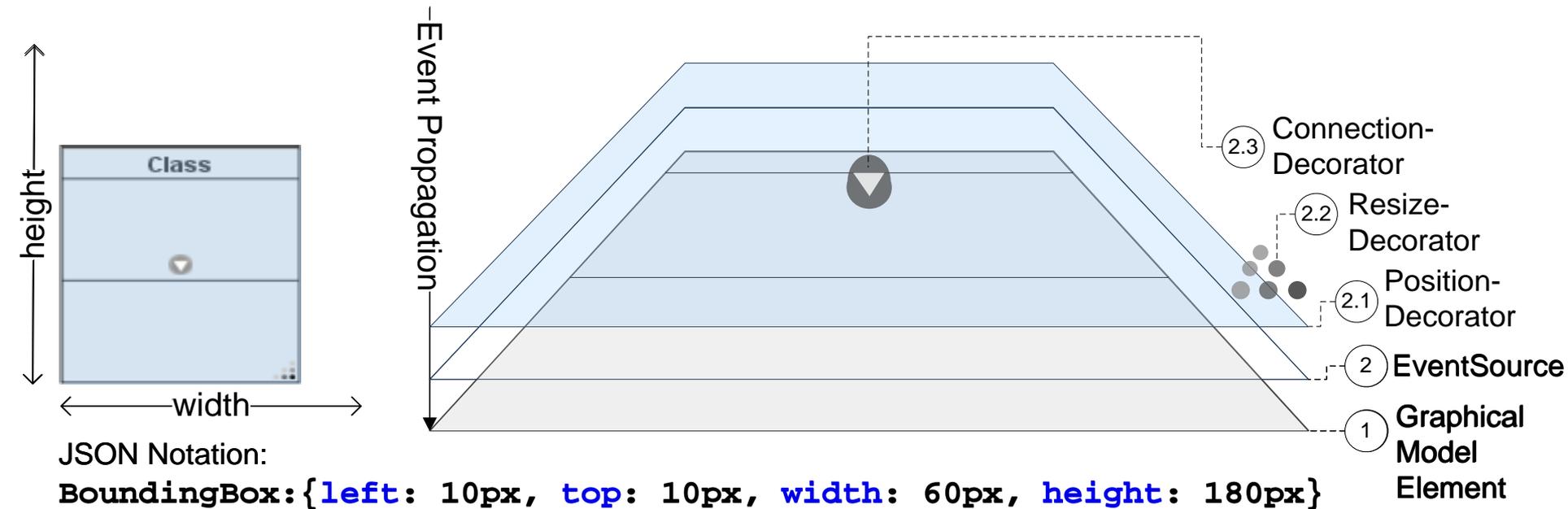
Security concerns
Deployment issues
Maintenance effort





The SLiM Collaboration Environment: Diagram Editor

- Challenge: Interacting with graphical elements in the browser



- Although SVG and VML generally provide interfaces for event handling, browser support is limited.



The SLiM Collaboration Environment: Event-notification via the HTTP protocol

- Challenge: Notifying clients about remote events
- The HTTP protocol, which is used as the transport layer for communication, is built upon the request/response paradigm. All communication has to be initiated by the client.
- Because of this limitation, the HTTP protocol was believed to be not suitable for implementing synchronous collaboration.

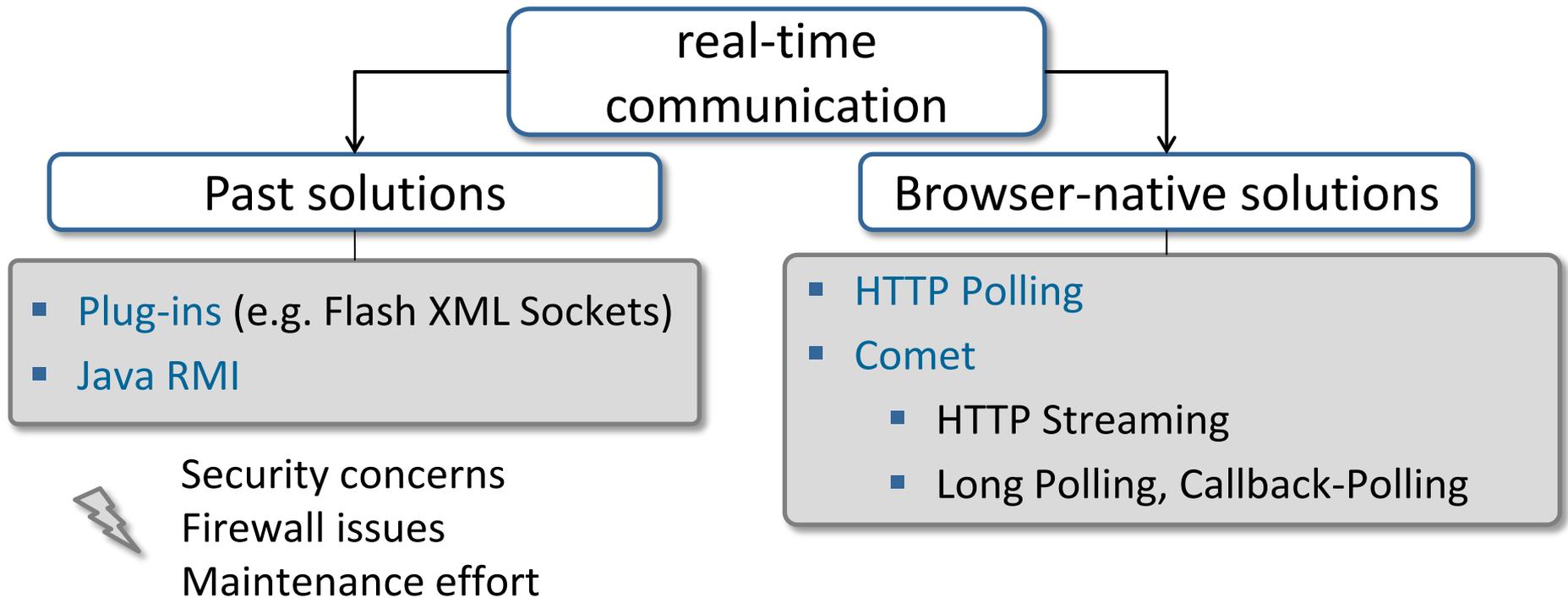
“HTTP is certainly **not suited** to real-time conversations”

[Dix 1997, Challenges for Cooperative Work on the Web]



The SLiM Collaboration Environment: Collaboration Support

- Challenge: Notifying clients about remote events



The SLiM Collaboration Environment: Comet - Scalable HTTP-based event routing

Client

Browser UI

User Activity

Comet Client

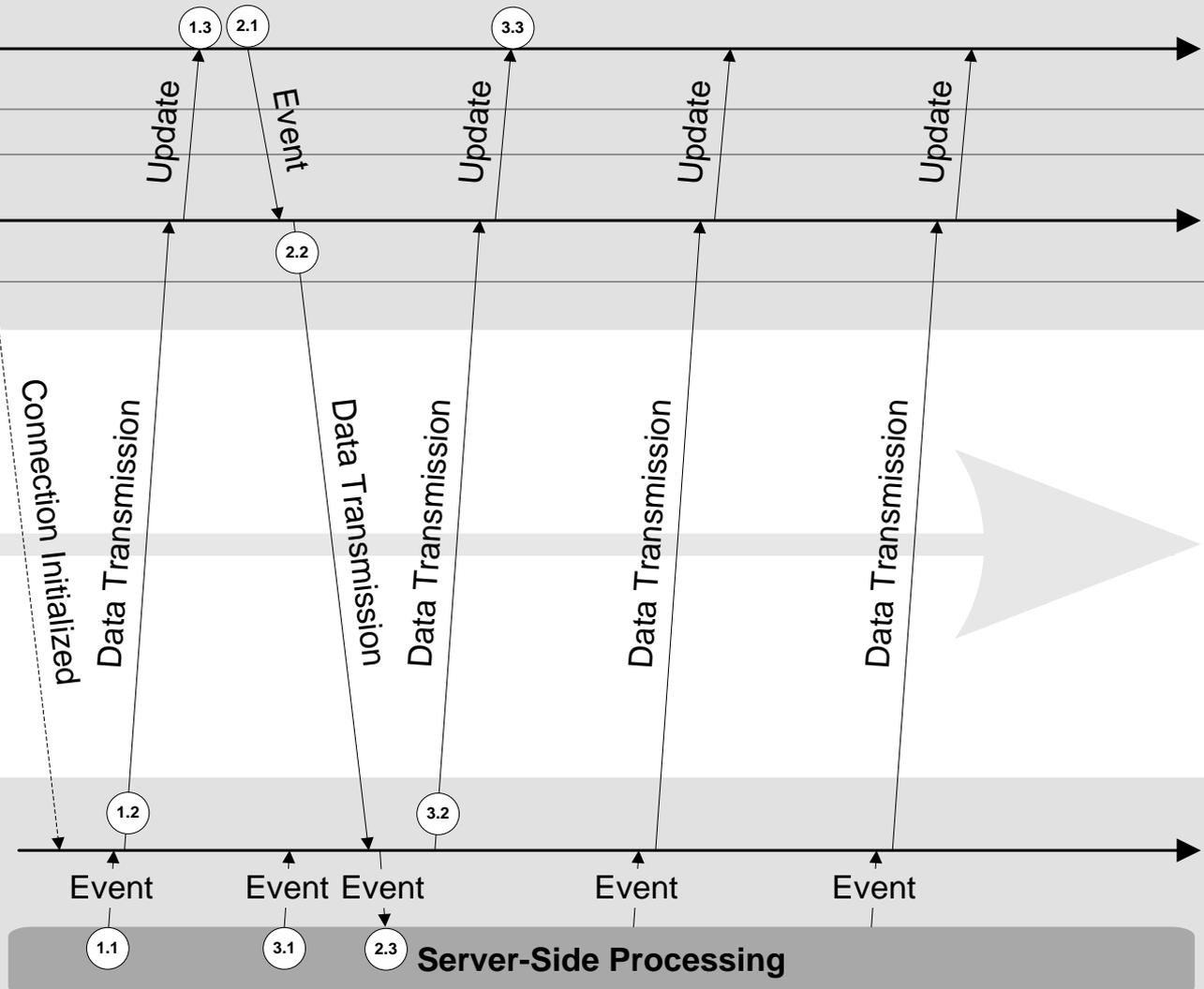
Client-Side Processing

Time (t)

Server

Comet Event Bus

Server-Side Processing



[based on Russell 2007]



Conclusion

- Technical feasibility of a lightweight browser-native tool that supports the modeling process in distributed software-engineering processes was shown
- By leveraging the benefits of web technologies, tools can be created, which considerably reduce the barriers of engaging in collaborative work
- In particular, the dependence of the distributed team members on the availability and capability of their IT department is reduced.



Future Research

- Future research will target several important aspects
 - a deeper insight into organizational and UML-related challenges in collaborative scenarios
 - Integration of SLiM with a traceability and versioning system
 - User-Study: SLiM in student team projects

**Further
Information** 

<http://www.wifo.uni-mannheim.de/~slim/>

Thank you for your attention!



**Any questions?
Feedback is appreciated.**

contact: thum@uni-mannheim.de

References

- Dix, A.: Challenges for cooperative work on the web: An analytical approach. *Computer Supported Cooperative Work (CSCW)* 6(2), 135–156 (1997)
- Russell, A.: Comet: Low latency data for the browser (2006), <http://alex.dojotoolkit.org/?p=545>
- Sengupta, B., Chandra, S., Sinha, V.: A Research Agenda for Distributed Software Development. In: *Proceedings of the 28th International Conference on Software Engineering (ICSE 06)*, Shanghai, China, pp. 731-740 (2006)
- Whitehead, J.: Collaboration in software engineering: A roadmap. In: *Future of Software Engineering (FOSE 2007)*, Minneapolis, Minnesota, USA, pp. 214–225 (2007)