Software Design:
Web-based Collaborative Development
Organization and Topics

Hasso-Plattner-Institut Potsdam
Software Architecture Group
Jens Lincke, Stefan Lehmann, Tim Felgentreff

http://www.hpi.uni-potsdam.de/swa/

2015/2016

What is Lively Kernel

- Web-based Development Environment
  - Browser as operating system
  - Interactive and explorative development of web applications
  - Wiki-like collaborative environment
- Self-supporting Development
  - No separate run-time and development environments
  - Tools and applications can be changed while being used
- Morphic
  - Direct manipulation of graphical objects
  - The UI is the application
  - Tools and applications can be explored and (de-)composed from the user's perspective

Demo of Lively Kernel

What is this Seminar about

- Lively Kernel: A web-based development environment

- Webwerkstatt (collaboration through one repository)

  vs.

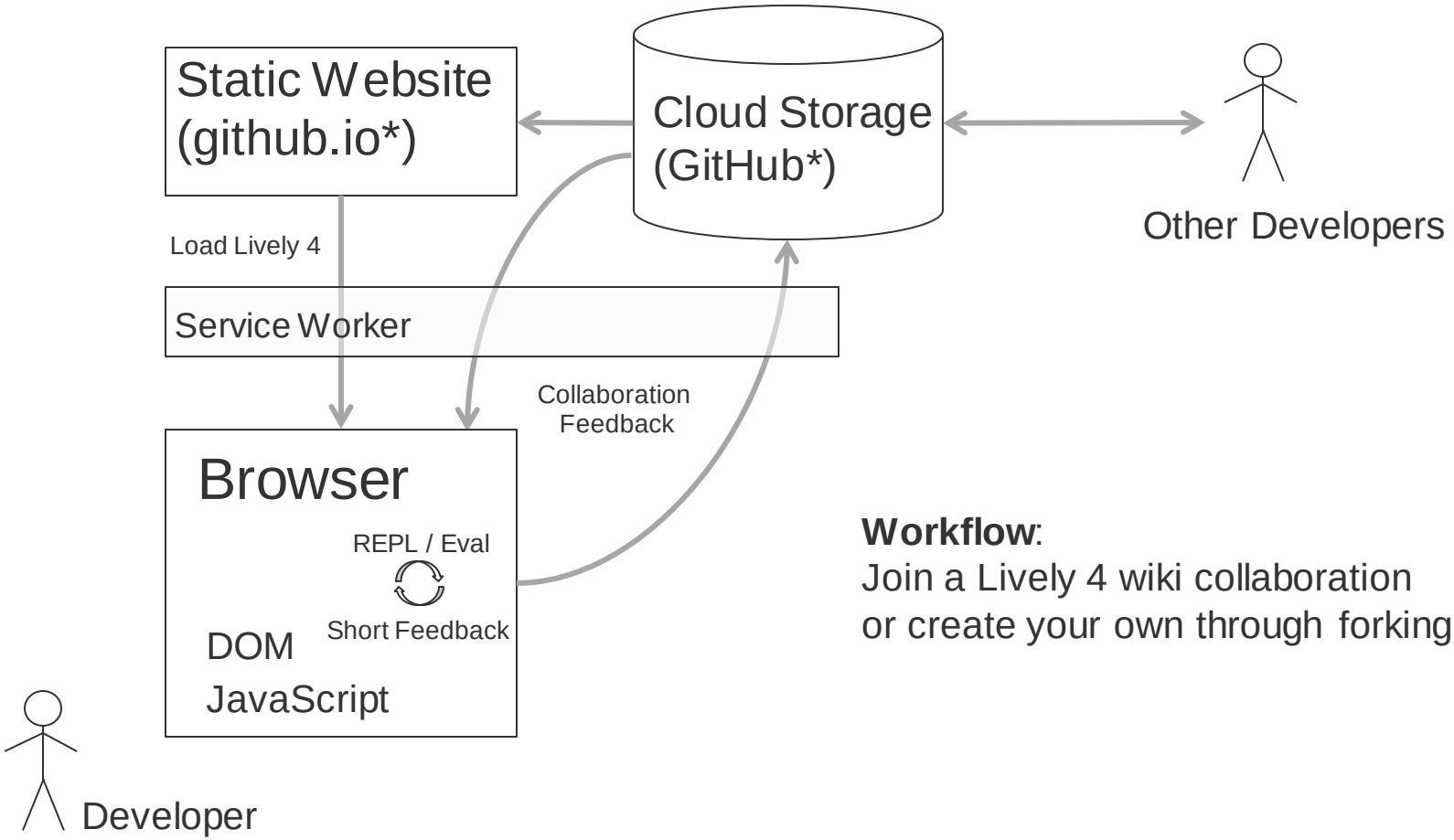
- GitHub-based Lively Kernel development (localhost)

- Goal: Lively 4
  - Web-based development
  - Collaboration through GitHub (incl. forking and branches)

Motivation of Lively4

- Lively Kernel as a module
  - Lively 4 should be better modularized
  - The kernel should be itself a module (and play nice…)
  - Use state of the art projects / modules (ACE editor, Babel)
- Embrace new Web-technology
  - User data is HTML5 (active and passive content)
  - Service Worker / Background Sync (offline first)
  - ES 2015 et al (source code rewriting)
- Collaboration
  - Lightweight and direct (no local git and node.js needed)
  - Async and sync
  - GitHub → Dropbox → Google Drive

Lively 4 Core Architecture



Workflow:
Join a Lively 4 wiki collaboration
or create your own through forking

* and others...

Demo of Core Architecture

Project 1: Thin Morphic on HTML5

- Background: Lively Kernel implements Morphic as object hierarchy that is rendered (HTML)

- Instead, make DOM-nodes the Morphic hierarchy
  - Halos
  - Inspection (State)
  - Scripts (Behavior)
  - Attribute Connections (Events / Behavior)

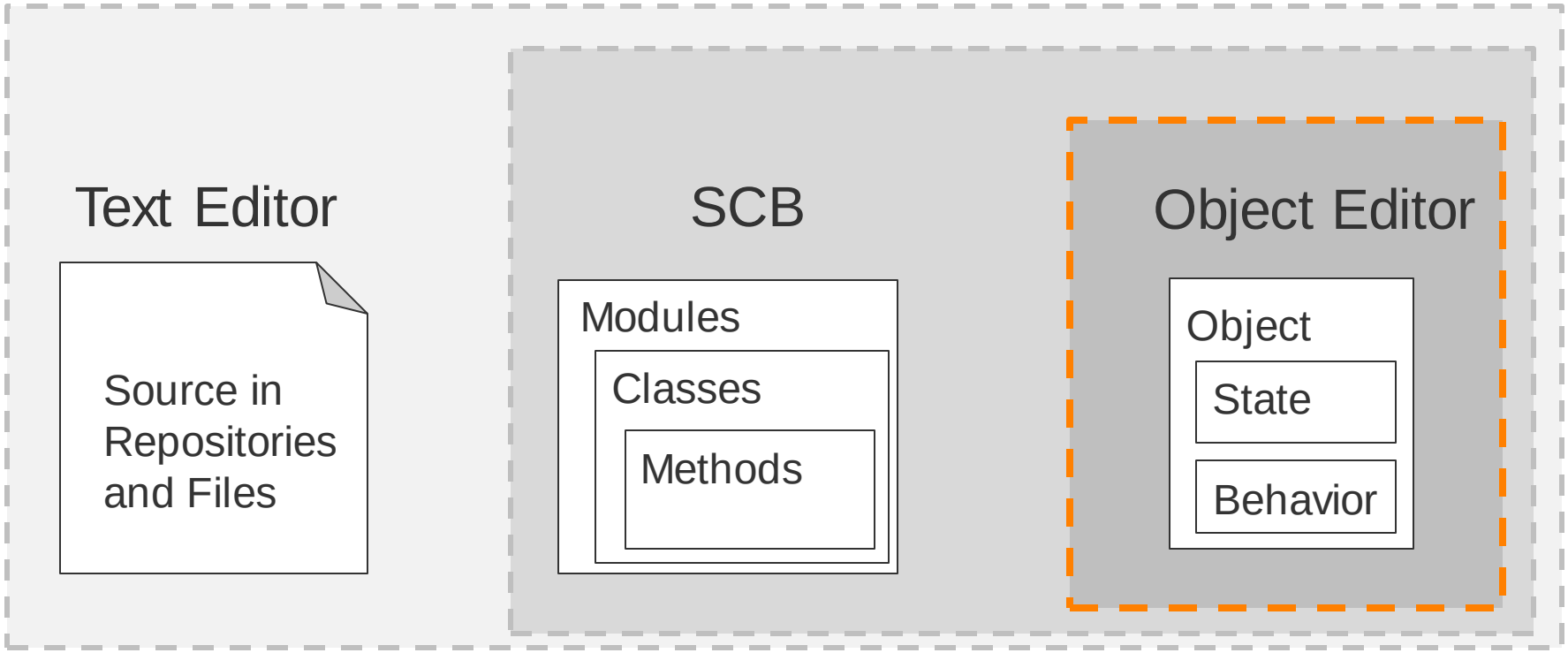
- Goal: Interactive editing of HTML pages

Project 2: Persistent HTML DOM

- Morphic on Shadow DOM
  - Full control over UI and User data changes
- State vs. History of changes
  - Continuously observe state changes (shadow DOM)
  - Static representation as readable HTML5 file
- Idea: Files in Git repository are snapshot of a history of changes. Both perspectives are useful.
- Goal: Get rid of the save button

Project 3: Live Object Editor

- Plain File Editor vs. System Code Brower (SCB)
- Editing files vs. behavior in objects
- Inspecting and editing state vs. behavior
- Goal: Live Object Editor for State and Behavior



Other Topics (not core, but nice to have)

- Reflective Rich Text / HTML Editor
- Remote Lively IDE (Lively 2 Lively)
- Build-in software visualization (get overview / stats)
- Unified backend API
  - GitHub, Dropbox, Browser Cache, File System, …
- Tile scripting environment
- Foreign modules: Smalltalk, Lisp….
- Touch and pen support

… and many more

Deliverables / Project Results

- Inter-team collaboration, because of dependencies!
  - Sorry, not just coding on the last weekend!
  - Code is immediately part of the project (Github)
- Explorative development
  - Developers are their own users (eat your own dogfood)
- Short weekly meetings (informal)
  - Feedback for helping project steering
  - Solving inter-team requirements

Organization

- Course
  - Weekly meetings (Slot to be found)
  - Project-Seminar, 4 SWS, 2-4 students per group
- Grading
  - 6 ECTS graded credit points
  - Grade based on project work and presentations
- Hand-In
  - Presentation, Screencast (HD), Source Code (MIT license)
- Important dates
  - Project topics on Oct 13th
  - Enrollment with prioritized topics on or before Oct 19st
    - Mail to jens.lincke@hpi.de
  - Topic assignment on Oct 20nd
  - Presentation dates determined after topics are assigned

Getting Started

- Become collaborator on https://github.com/LivelyKernel/lively4-core
  - We invite you after receiving your user name
- Fork your own wiki as needed, but be prepared to merge
- Goal: develop from within the browser, but fall back to other means as needed during bootstrap phase
- Start building your project
- Make use of interactive development, but also write tests for Travis CI
- Ask questions and ask for help! https://github.com/LivelyKernel/Lively4/wiki/SWD15

Summary

- Mail to jens.lincke@hpi.de as a team with topics
  - Form teams (2–4 pers)
  - Prioritize the three core topics
  - Choose one of “Other Topics” as backup
- Co-develop the projects during the semester
  - Weekly meetings
  - Continues collaboration on GitHub
- Presentation at the end of the semester

Info Lively Kernel

Repos:

- github.com/LivelyKernel

Online demos:

- lively-kernel.org/
- lively-web.org/