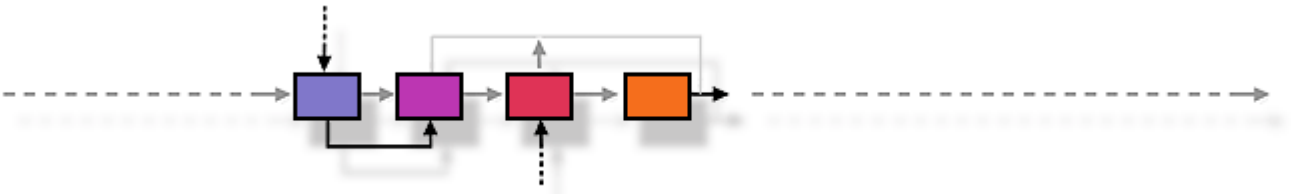




Java  
Community  
Process



## JSR 377

# Desktop Application Framework

September 29 2017

Andres Almiray

# Agenda



- Goals
- Information to be gathered
- Implementation notes
- Issues
- Questions, discussion, next steps



# Goals

# Goals



- Define APIs that can be used to build Java Desktop (and possible mobile) applications.
- Inspired in previous efforts such as JSR 296 (Swing Application Framework) and JSR 295 (Beans Binding)
- JSR 377 does not target a single UI toolkit. It should be possible to use existing toolkits such as Swing, JavaFX, SWT, Apache Pivot, Lanterna, others.



Information to be gathered

# About this JSR



- Provide the following features
  - application life cycle
  - localized resources (and injection)
  - persisted session state
  - loosely coupled actions
  - dependency injection
  - event system
  - centralized error management
  - extension points via plugins

# Introduction



- JSR 193 – Client Side Container
- JSR 296 – Swing Application Framework
- JSR 295 – Beans Binding
  
- JSR 296 had the following goals
  - application life cycle
  - localized resources (and injection)
  - persisted session state
  - loosely coupled actions
  - targeted only Swing for obvious reasons

# Business/marketing/ecosystem justification



- Why do this JSR?
  - Desktop application development can benefit from lessons learned by JavaEE standardization.
- What's the need?
  - Applications can't share code/components unless they are written using the same base platform (NetBeans, RCP, Eclipse RCP, homegrown).
- How does it fit in to the Java ecosystem?
  - Desktop applications are still being built.
- Is the idea ready for standardization?
  - UI toolkits and desktop platforms are mature enough.



# History



- List the significant dates in the history of the JSR.
  - Submitted on 29 Dec, 2014.
  - Review Ballots: 27 Jan, 2015  
16 Feb, 2016  
14 Mar, 2017
  - Preliminary API drafted.
  - Spec document not ready.
  - Potential RI identified.

# Technical scope and features



- API targets JDK7 byte code due to restrictions found in mobile platforms. This decision was taken back in 2014 and reviewed in following years. It's very likely the new target can be set to JDK8.
- The API is independent of any UI toolkit at its core but provides hooks for specific implementation, such as the case of handling the particulars of the UI thread.
- All original features proposed by JSR-296 are in scope.
- JSR-295 has been superseded by JavaFx bindings.

# The Expert Group



- The EG is comprised by Java Champions, JavaFX experts, NetBeans Dream Team member, and other interested parties.
- The EG discusses ideas using the mailing list, GitHub issues, Gitter (online chat) and occasional face-to-face meetings at conferences and JUG events.

# Other deliverables



- Other than Spec, RI, and TCK are you delivering, for example:
  - Additional documentation
  - User's guide
  - Sample code

# Publicity



- <https://jug.bg/ftp/jPrime/2016/slides/JSR377:%20What's%20up%20and%20what's%20next%20-%20Andres%20Almirai.pdf>

# Implementations



- How many implementations (apart from the RI) exist?
  - None yet

# Schedule



- Indicate the expected schedule to completion of the JSR, with milestones and JSR stages.

# IP flow



- Provide pointers to the licenses for the Spec, RI, and TCK.
  - Apache v2 License
- How are you handling contributions from non JCP members?
  - Apache v2 License
- What Terms of Use apply to your collaboration tools?
  - None
- Do you have a Contributor Agreement?
  - Yes
- Any legal issues or concerns?
  - None at the moment



# RI and TCK development



- How are you developing the RI and TCK?
  - In the open using the Griffon framework as testbed
- If collaboratively (through an open-source project)
  - How many committers and who?
    - Andres Almiray, Alexander Klein
- Is the RI available for public download?
  - <http://github.com/griffon/griffon> (not yet retrofitted with jsr377-api)
- Is the TCK available for public download?
  - Not yet
- Do you have a source-code repository?
  - <https://github.com/jsr377>

# Participation and transparency



- Provide a pointer to the JSR page on JCP.org
  - <https://jcp.org/en/jsr/detail?id=377>

# Adopt-a-JSR



- Are you participating in the Adopt-a-JSR program?
  - Not yet.
- If so, what kind of contributions/help have you received and from whom/which JUGs?
  - We have received a handful of contributions at Hackergarten meetings in Basel, Zürich, Luzern, and JavaOne.

# Mailing lists or forums



- How are you communicating with the public and how can they communicate with you?
  - <http://jsr377.github.io/site/>
  - <https://twitter.com/jsr377>
- Provide pointers to public mailing list(s) and/or forum(s)
  - <http://jsr377-api.40747.n7.nabble.com/>
- Total number of participants (EG members, non-EG members?)
  - 20

# Issue tracker



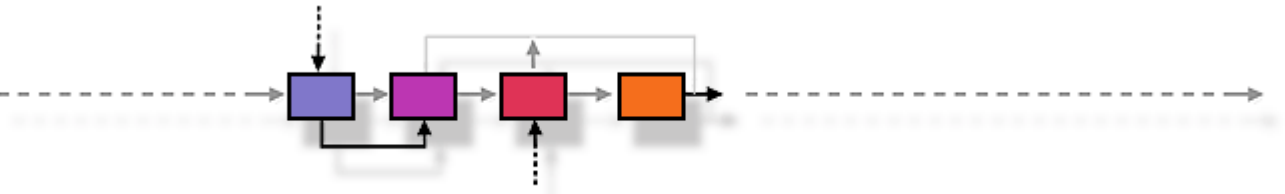
- Total number of issues?
  - 2
- How many in each state (open, closed, deferred, etc?)
  - 2 open
- Average number of issues logged per month?
  - 0
- Average number of issues resolved per month?
  - 0
- How many different people logged them?
  - 1



Questions, discussion, next steps



Java  
Community  
Process



Thank you!  
<http://jcp.org>