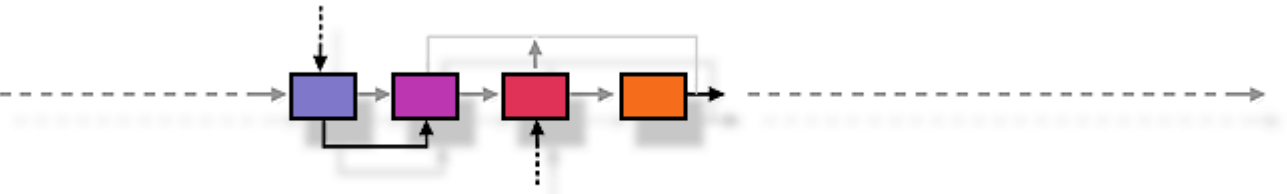# JSR 367 (JSON Binding) Review
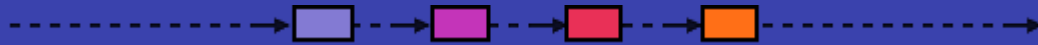
September 15 2016

Dmitry Kornilov

Agenda

- Goals
- Information to be gathered
- Next Steps
- Issues
- Q&A

Goals

Goals (1/2)

- Support binding (serialization and deserialization) for all RFC 7159-compatible JSON documents.
- JSON-related specifications will be surveyed to determine their relationship to JSON-Binding.
- Maintain consistency with JAXB and other Java EE and SE APIs where appropriate.
- Define default mapping of Java classes and instances to JSON document counterparts.
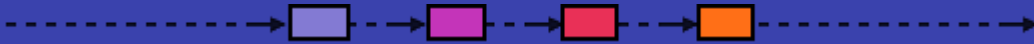- Allow customization of the default mapping definition.

Goals (2/2)
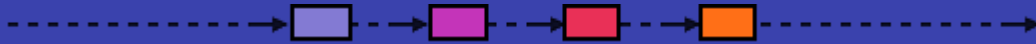
- Default use of the APIs should not require prior knowledge of the JSON document format and specification.

- Define or enable integration with JSR 374: Java API for JSON Processing (JSON-P) 1.1.
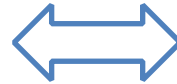
Information to be gathered

JSON-B is standard binding layer for converting Java objects to/from JSON documents

Java

```
public class Customer {
    public int id = 1;
    public String firstName = "John";
    public String lastName = "Doe";
    ….
}
```
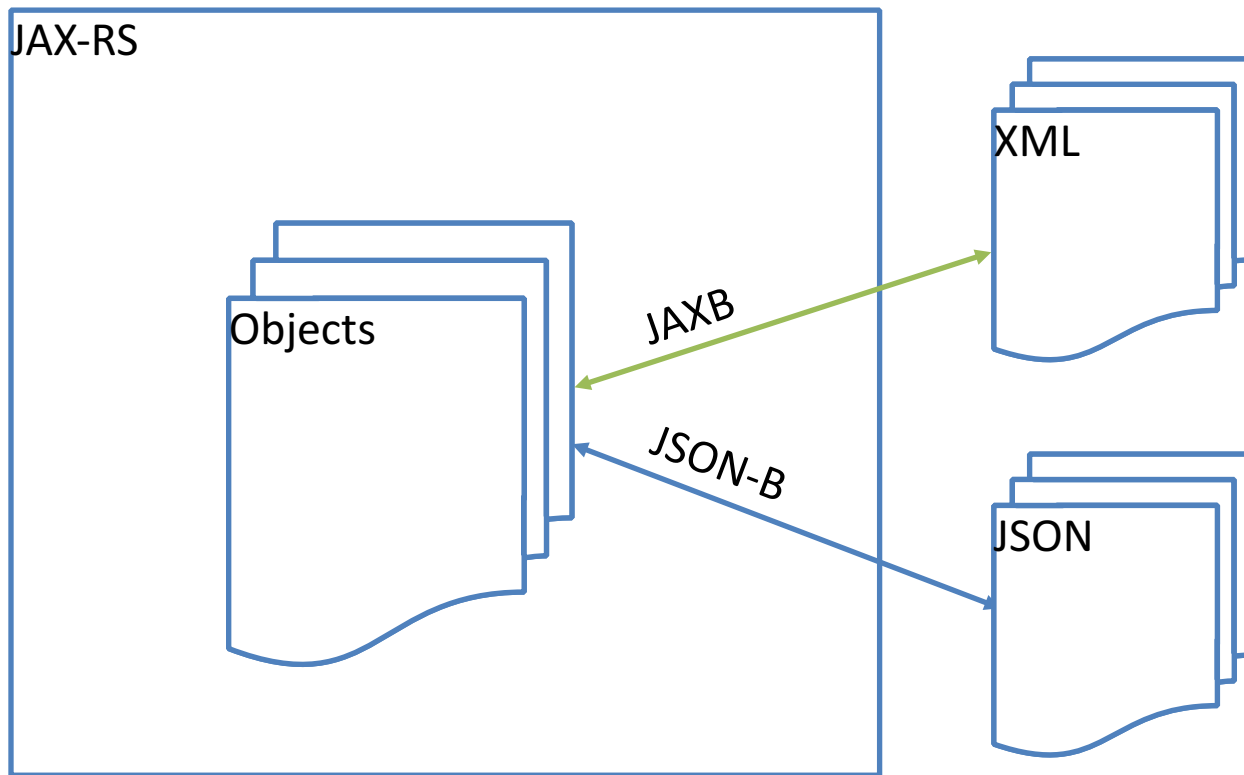
JSON

```
{
    "id": 1,
    "firstName" : "John",
    "lastName" : "Doe",
}
```

Java Community Process

Usage in Other Java EE Frameworks

JAX-RS

Objects

JAXB

JSON-B

XML

JSON

Introduction

- Supports Java EE 7
- Targeted for inclusion in Java EE 8
- One of the core parts of Java EE 9

Which of these APIs do you think is important to be included into Java EE 8?

	Yes	No	Not sure (%)
JCACHE (JSR 107)	67.4	6.3	26.3
Data Grid API (JSR 347)	41.6	14.3	44.1
State Management API (JSR 350)	36.7	12.3	51
Identity API (JSR 351)	47.2	9.5	43.3
Java API for JSON Binding	78.8	5.6	15.6
Java EE Configuration	63.7	6.9	29.4

0   25   50   75   100

Java Community Process

10
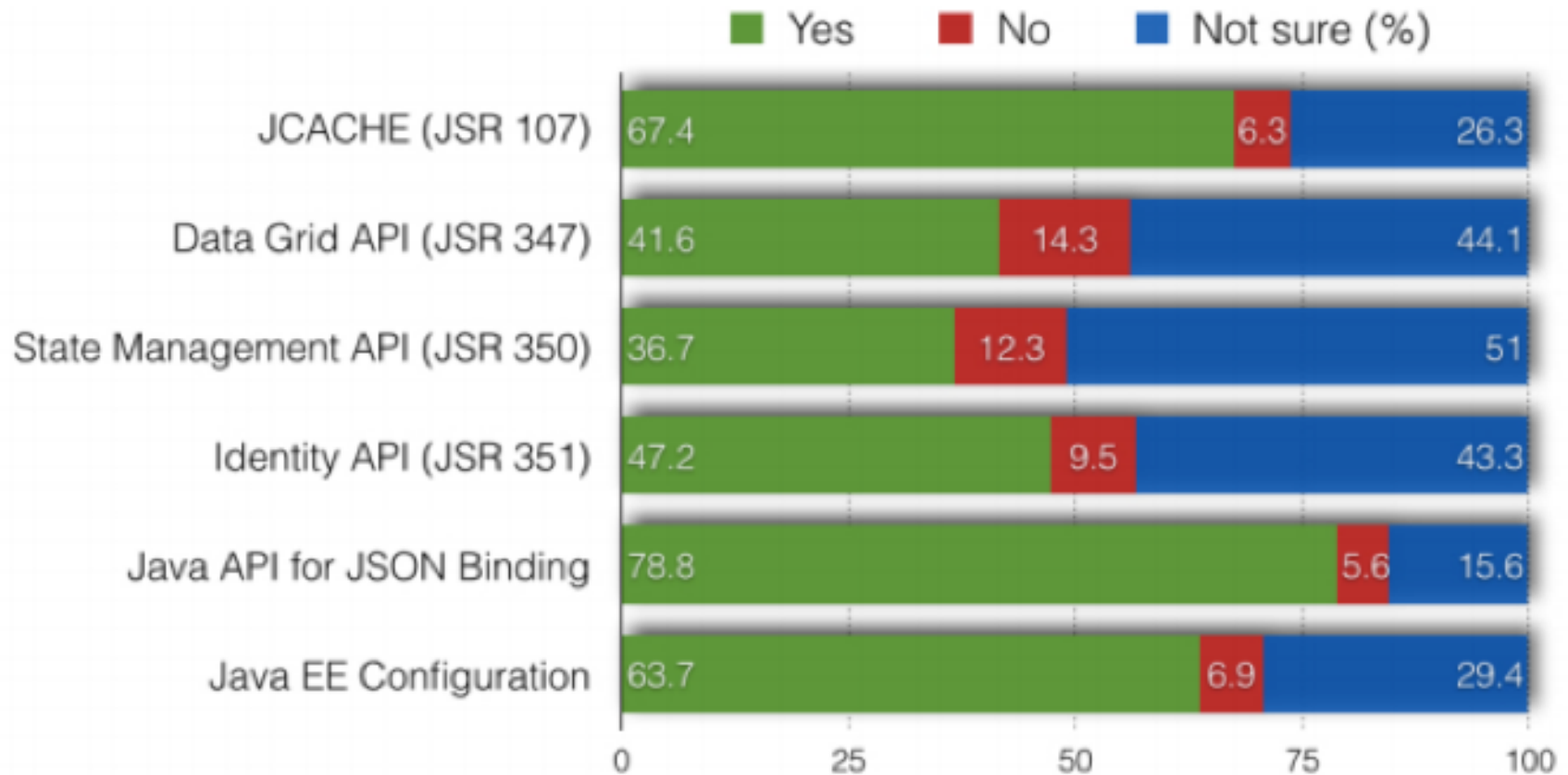
Business/marketing/ecosystem justification

- Many services are offering data via JSON messages that Java developers need to interact with as Java objects.

- There is a need in other frameworks like JAX-RS to convert objects to/from JSON

- JAXB analog for JSON

- Standardize current technology

History

- 26 Aug 2014 – JSR Submitted
- 23 Sep 2014 – Expert Group Formation
- 20 Aug 2015 – Early Draft Review
- 26 July 2016 – Public Draft Ballot

Technical scope and features

Default Mapping

– Basic Types
– Specific Types
– Dates
– Classes
– Collections/Arrays
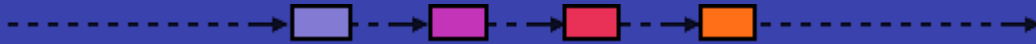– Enumerations
– JSON-P types

Customized Mapping

– Property names
– Property order
– Ignoring properties
– Null handling
– Custom instantiation
– Custom visibility
– Date/Number Formats
– Binary Handling
– Adapters
– Serializers/Deserializers

Technical scope and features

- Annotation based customizations
- Easy runtime customizations using JsonbBuilder
- Full support of generic types
- Integration with JSON-P
- I-JSON support

The Expert Group

- **Dmitry Kornilov (Oracle)**
- Przemyslaw Bielicki
- Eugen Cepoi
- Martin Vojtek (Datlowe)
- Rick Curtis (IBM)
- Nathan Rauh (IBM)
- Roman Grigoriadi (Oracle)

- Alexander Salvanos
- Hendrik Saly
- Otavio Santana
- Inderjeet Singh
- Kyung Koo Yoon (TmaxSoft)
- Romain Manni-Bucau (Tomitribe)
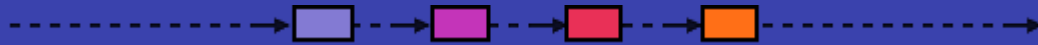- Gregor Zurowski

The Expert Group

- Main communication channel is experts mailing list
- If needed we setup a videoconference using Zoom, Skype or Hangouts
- Face to face conversations

Other deliverables

- JSONB web site (jsonb.java.net)
- Samples is a part of the spec
- Sample projects on GitHub

Publicity

- JavaOne 2015
- EclipseCon 2016 NA
- JavaOne 2016

Collaboration with other community groups

- Presentation for CZ JUG

- Presentation for Bentonville JUG

- Faso JUG Adopts JSR 367
    - https://github.com/pandaconstantin/adopjsrfasojug

- First look on JSR-367 (Turkish)
    - https://blog.kodcu.com/2016/09/json-binding-jsr-367-incelemesi/

- JSONB test-drive using JAX-RS
    - https://abhirockzz.wordpress.com/2016/08/24/json-b-test-drive-using-jax-rs/

IP flow

- Spec, RI and TCK licenses:
  - https://www.jcp.org/en/jsr/detail?id=367
- Code contributions from non JCP members:
  - Accepted on GitHub or Eclipse.org
  - Reviewed and merged
  - We didn't have any yet
- Terms of Use of our collaboration tools:
  - https://www.java.net/javanet-web-site-terms-use
  - https://help.github.com/articles/github-terms-of-service
- No legal issues so far

RI and TCK development

- RI is developed as an open-source project
  - http://git.eclipse.org/gitroot/eclipselink/eclipselink.runtime.git
- Binary download
  - https://oss.sonatype.org/content/repositories/snapshots/org/eclipse/persistence/jsonb-ri/1.0-SNAPSHOT/
- Main committers:
  - Roman Grigoriadi (Oracle)
  - Dmitry Kornilov (Oracle)
  - David Kral (Oracle)
- TCK development is not started yet

Participation and transparency

- JCP.org page
  - https://www.jcp.org/en/jsr/detail?id=367
- Specification project page
  - https://jsonb-spec.java.net

Adopt-a-JSR

- We are participating in Adopt-a-JSR program
  - Presentation for CZ JUG
  - Presentation for Bentonville JUG
  - Faso JUG Adopts JSR 367
    - https://github.com/pandaconstantin/adopjsrfasojug
- Where is no contribution to the project received

Mailing lists or forums

- Mailing lists:
  - Experts, Users, Issues
  - https://java.net/projects/jsonb-spec/lists
- Total number of messages: 562
- Average per Month: 25.5

Mailing lists or forums

Mailing lists messages
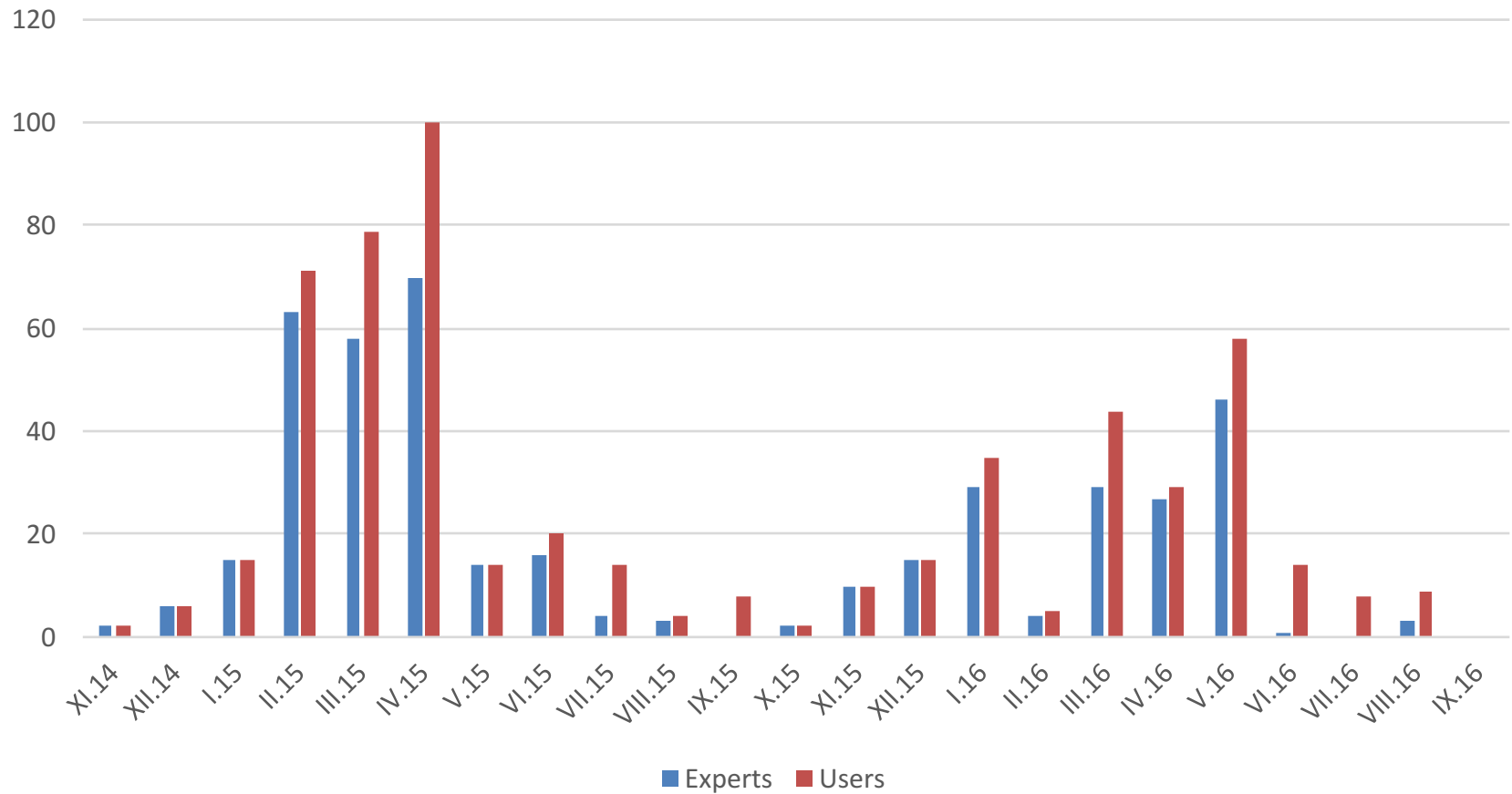
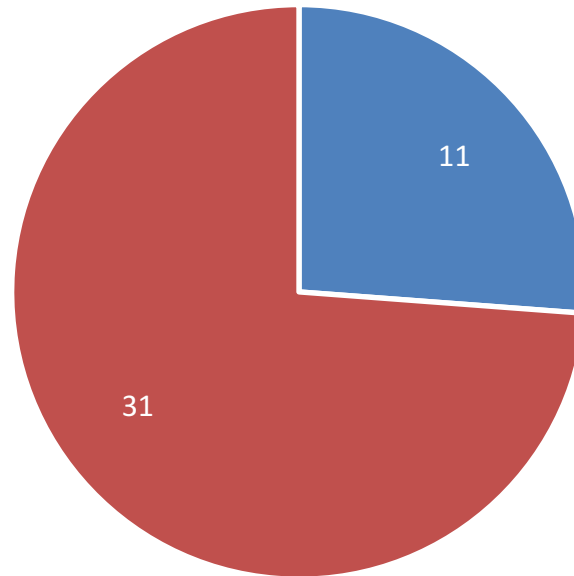# Mailing lists or forums

Emails in mailing lists by type
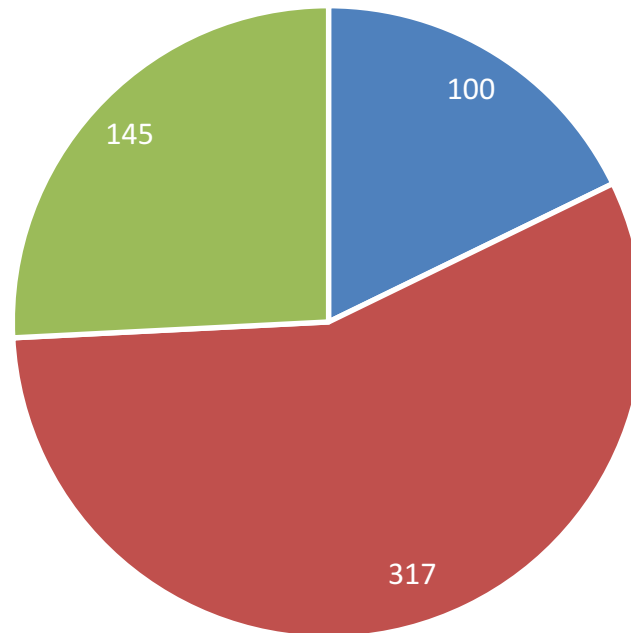
- Spec Lead
- Experts
- Users

(100, 145, 317)

Issue tracker
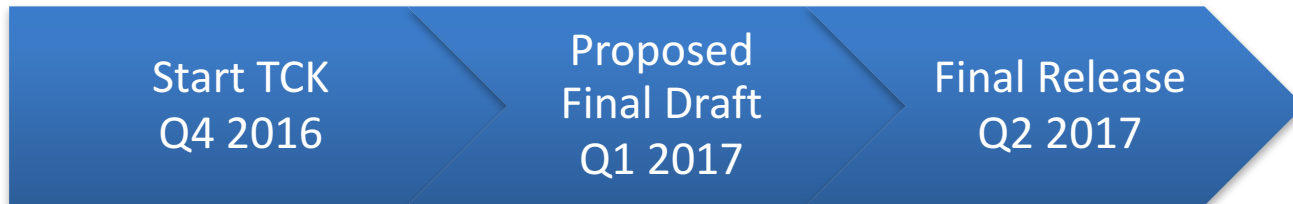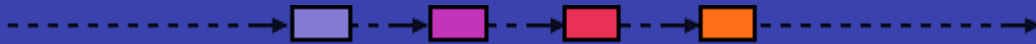
- http://java.net/jira/browse/JSONB_SPEC

Issues by status

Open Closed

Next Steps

Schedule

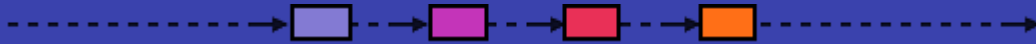Start TCK
Q4 2016

Proposed
Final Draft
Q1 2017

Final Release
Q2 2017

Next Steps

- RI
  - Performance testing
  - Performance comparison
  - Performance optimization
  - Test on real life use cases
- TCK
- Evangelism
  - Samples, guides, manuals
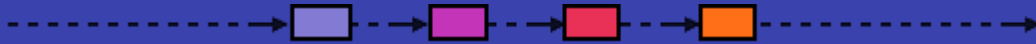 - Blog articles
  - Conferences

Next Steps

- JSONB 1.1
  - Integration with other Java EE frameworks (JAX-RS, JPA)
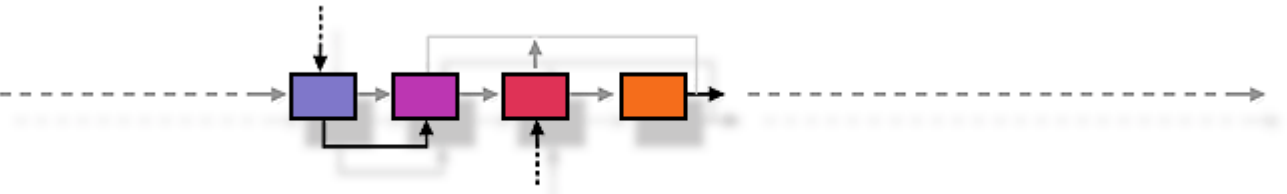 - JSON Pointer
  - Partial Mapping

Issues

Issues

- Integration with JSON-P
- Resources for TCK

Questions?

Thank you!
http://jcp.org