

Whats coming in Java EE 8?

Arun Gupta, Red Hat
@arungupta



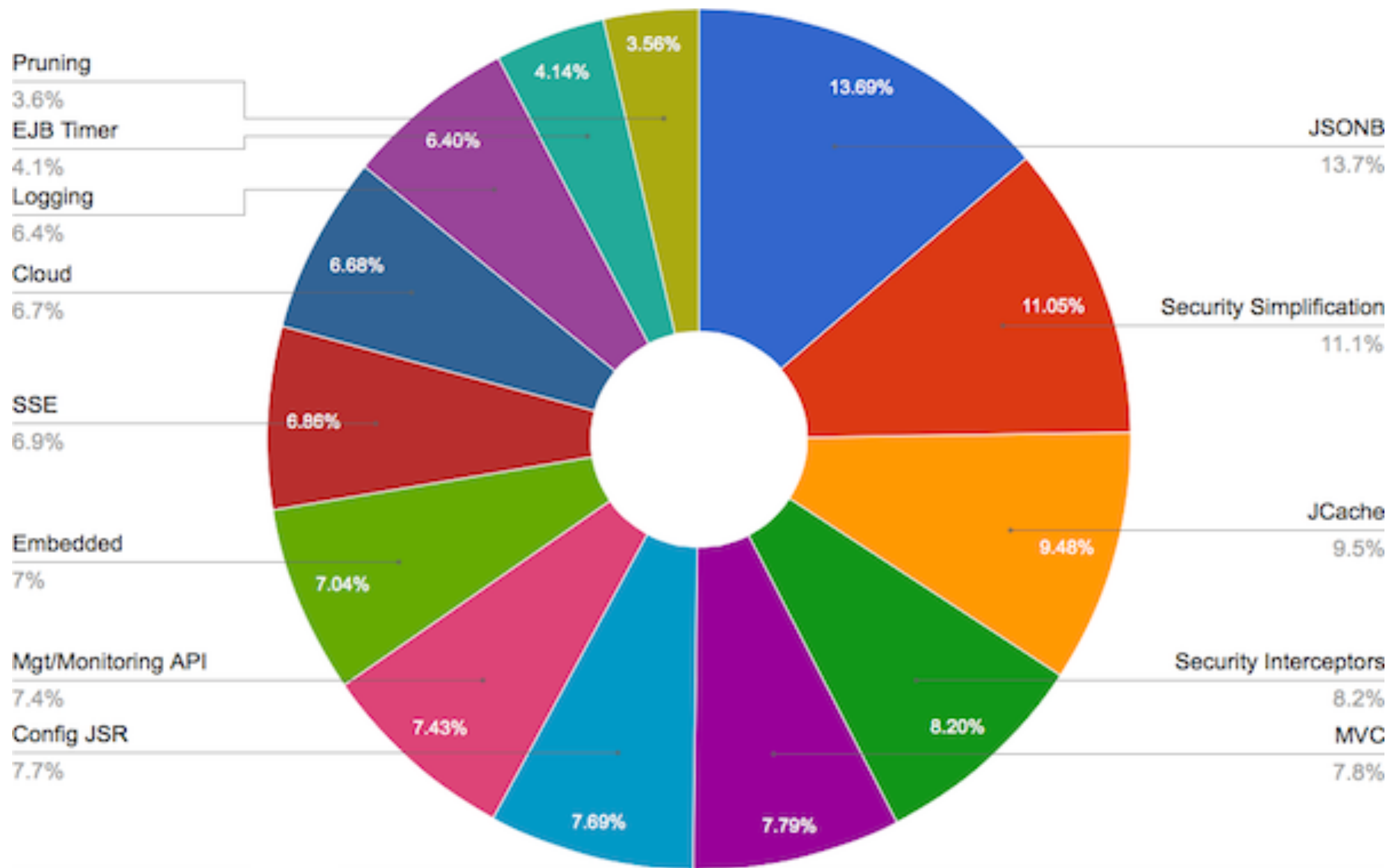
- More annotated POJOs
- Less boilerplate code
- Cohesive integrated platform

- WebSockets
- JSON
- Servlet 3.1 NIO
- REST

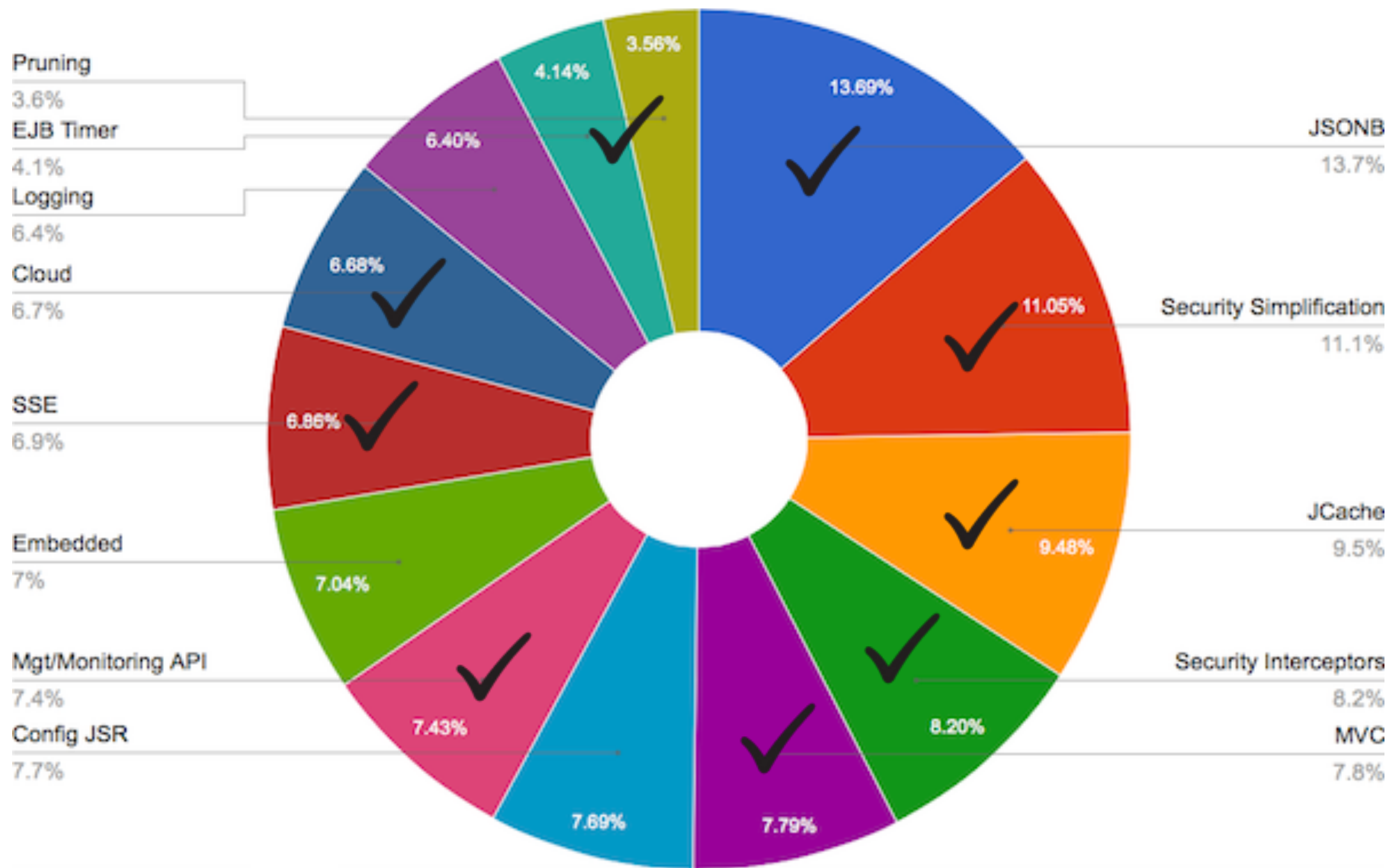
- Batch
- Concurrency
- Simplified JMS

Top 10 features

- WebSocket client/server endpoints
- Batch Applications
- JSON Processing
- Concurrency Utilities
- Simplified JMS API
- Transactions in POJO
- JAX-RS Client API
- Default Resources
- More annotated POJOs
- Faces Flow



https://java.net/downloads/javaee-spec/JavaEE8_Community_Survey_Results.pdf



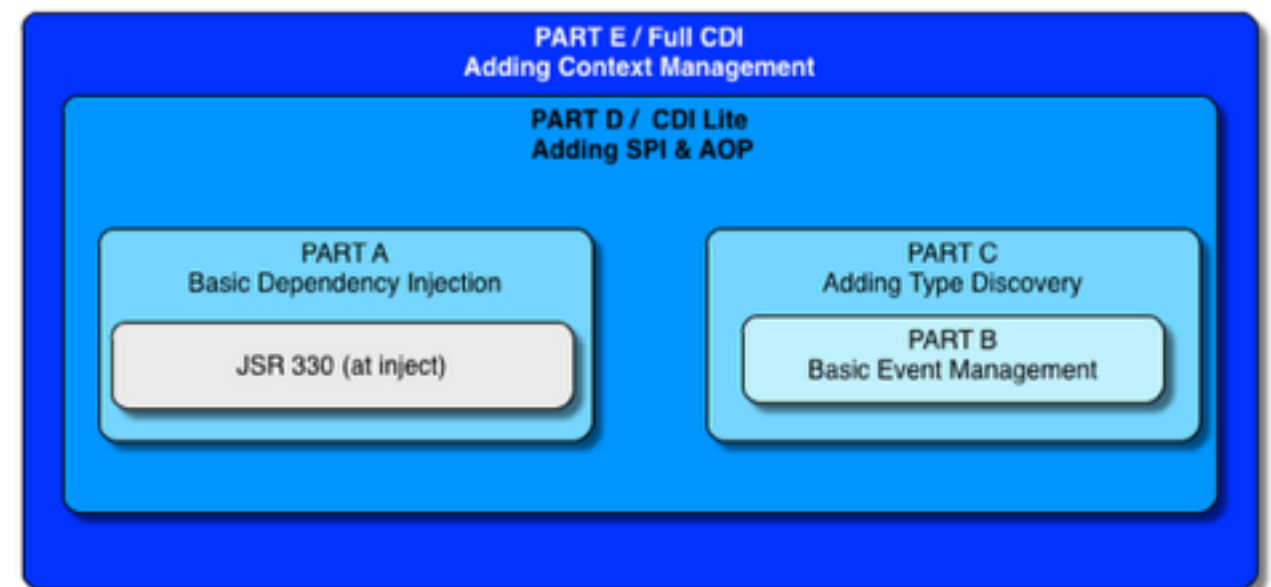
Component	JSR
Java EE 8	366
CDI 2	365
MVC 1.0	371
Servlet 4.0	369
JSON-B 1.0	367
Security 1.0	375
Management 2.0	373
JAX-RS 2.1	370
JMS 2.1	368
JSF 2.3	372
JSON-P 1.1	374
WebSocket 1.1	356
JCache 1.1	
Concurrency 1.1	
JPA 2.2	

Java EE 8 (JSR 366)

- Web Standards/HTML 5 Alignment
 - HTTP/2, SSE, JSON-B, Action-oriented framework, hypermedia
- CDI Alignment
 - CDI 2, EJB services outside EJB, Pruning
- Cloud
 - Simple security providers, multi tenancy, REST management/monitoring
- Enterprise
 - JCache, Configuration, JMS
- Java SE 8 Alignment

CDI 2 (JSR 365)

- Split CDI into two parts: Core and Java EE integration
- Portable bootstrap API for CDI
 - Enables Java SE bootstrap
- Asynchronous events
- Modularity
- Lightweight container



MVC 1.0 (371)

- Action-based MVC (JSF is Component-based)
- Model = CDI, Bean Validation, JPA
- View = JSPs or Facelets
- Controller = JAX-RS based
- No new templating language, SPI for integration

MVC 1.0 Code Sample

```
@Path("/")
@Template("my-index.xhtml")
public class Bookstore {
    ...
    @GET
    public List<Item> getItems() {
        ...
        return items;
    }
}
```

Servlet 4.0 (369)

- HTTP/2
 - Request/response multiplexing over single connection
 - Multiple streams
 - Stream prioritization
 - Server push
 - Binary framing
 - Header compression



JSON-B 1.0 (367)

- Marshal/unmarshal POJOs to/from JSON (~JAXB)
- Default mapping of classes to JSON
 - `@JsonProperty`, `@JsonValue`, `@JsonTransient`
- Lessons from existing solutions
 - MOXy, Jackson, GSON, Genson, ...
 - Pluggable providers
- Standard mapping in JAX-RS (currently only JSON-P)

Security 1.0 (375)

- Simplify and improve portability
- Simple security providers
- Enable existing annotations (`@RolesAllowed`) for all beans
- EL enabled security annotations via Interceptors
- Password aliasing
- OAuth/OpenID

Management 2.0 (373)

- Revamp dated JSR 77 (J2EE Management)
- REST URI for managed objects, CRUD operations
 - Instead of EJB 2.x remoting
- SSE/WebSocket for notification
- Deployment to be included
- Application deployment, resource definition, and management using the same interface

JAX-RS 2.1 (370)

- Server-Sent Events
- Improved integration with CDI
- Non-blocking I/O in providers(filters, interceptors, etc)
- Simplified hypermedia API
- Improve asynchronous API to simplify reactive programming

JMS 2.1 (368)

- Continuation of JMS2
- Declarative message listeners
 - Alternative to MDB
 - Available to all beans
- Improving JMS provider portability
- Minor features and corrections
 - Redelivery delay, delivery limits, dead message queues

JSF 2.3 (372)

- Invoke CDI methods from Ajax calls
- Community input
 - Multi-field validation
 - `@Inject FacesContext`
 - EL performance optimizations
 - Cross-form Ajax clarifications

JSON-P 1.1 (374)

- Adapt to new JSON standards
 - JSON-Pointer (IETF RFC 6901) - identify specific value within JSON document
 - JSON-Patch (IETF RFC 6902) - add, replace, remove, move, copy
- Editing operations for `JsonObject` and `JsonArray`
- Helper classes and methods to better utilize Java SE 8's stream operations

WebSocket 1.1 (356)

- `Session.addMessageHandler(MessageHandler)`
now work with Lambda expressions

Adopting Java SE 8

- Most of Java SE 8 can already be used with Java EE
- Some APIs can adopt features
 - Repeatabe annotations
 - Date-Time API/JDBC 4.2
 - Completable Future
 - Lambda expressions and streams
 - Default methods

Schedule

- Early Drafts: 2015
- Proposed Final Draft: Early 2016
- Final Drafts: Late 2016
- Reference Implementation: GlassFish 5.0

Latest Status: blog.arungupta.me/javaee8