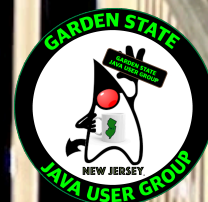


Jakarta EE 11: Going Beyond the Era of Java EE



JAKARTA® EE

Michael P. Redlich | @mpredli



InfoQ



Inspiration

- Jakarta EE 11: The First Big Leap for Jakarta
 - Steve Millidge, CEO at Payara
 - March 30, 2023



Steve Said...

“From the initial lift and shift to the new namespace change in Jakarta EE 9, to the simplification work done in Jakarta EE 10, a lot of effort has gone into making Jakarta EE a solid basis for open source developers to build on.”



Steve Also Said...

“With that done, there’s now the opportunity to start taking Jakarta EE beyond the Java EE era. With Java 21 [on the horizon], there’s now the opportunity to make sure Jakarta EE is always leveraging the latest and greatest capabilities of the new Java version, build new specifications and further unify and simplify the platform.”

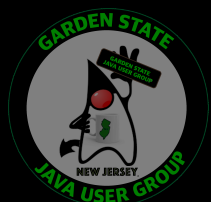




Hi, I'm Mike



- Java Champion
- Lead Java Queue News Editor: InfoQ
- Contract Dev Advocate and Tech Writer: Payara
- Director: Garden State Java User Group
- Leadership Council: Jakarta EE Ambassadors
- Committer: Jakarta NoSQL and Jakarta Data
- Senior Research Technician: ExxonMobil Technology & Engineering (retired with 33½ years service)



InfoQ



Objectives

- Evolution of Java EE/Jakarta EE
- Jakarta EE Profiles
- What's New in Jakarta EE 11
- Updated Specifications in Jakarta EE 11
- Highlighted Specifications with Demos



Poll

- I'm a Jakarta EE Expert!
- I've built an application using Jakarta EE
- I'm familiar with Jakarta EE, but haven't used it
- I've heard some things about Jakarta EE
- What the heck is Jakarta EE?



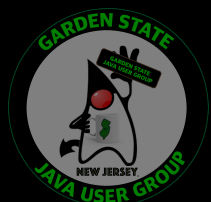
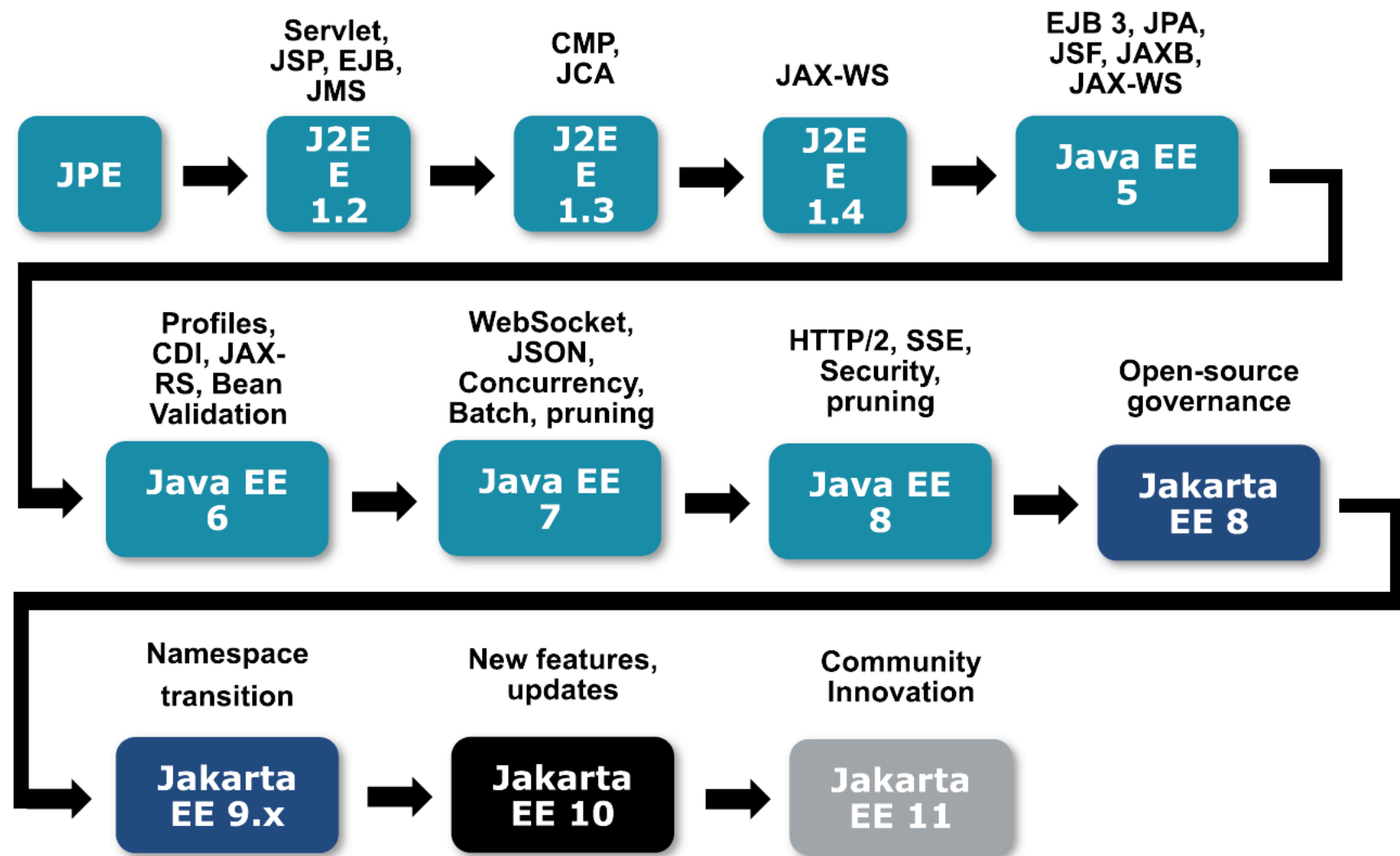
Evolution of Java EE/ Jakarta EE



InfoQ



Jakarta EE Evolution

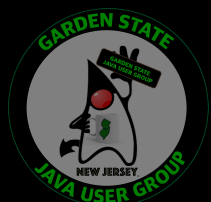
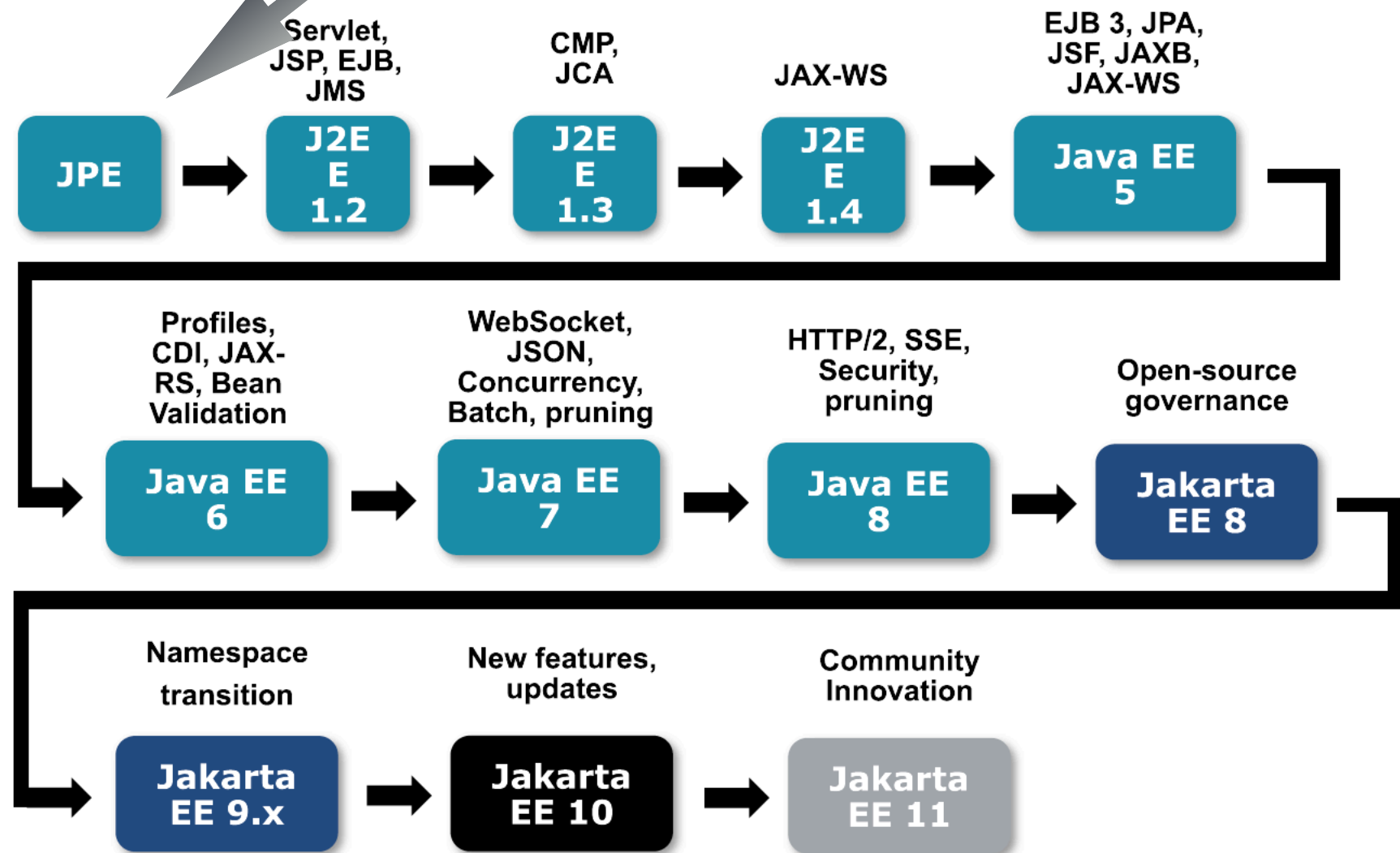


Java Professional Edition

Jakarta EE Evolution



JAKARTA® EE



InfoQ



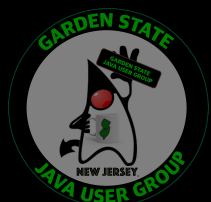
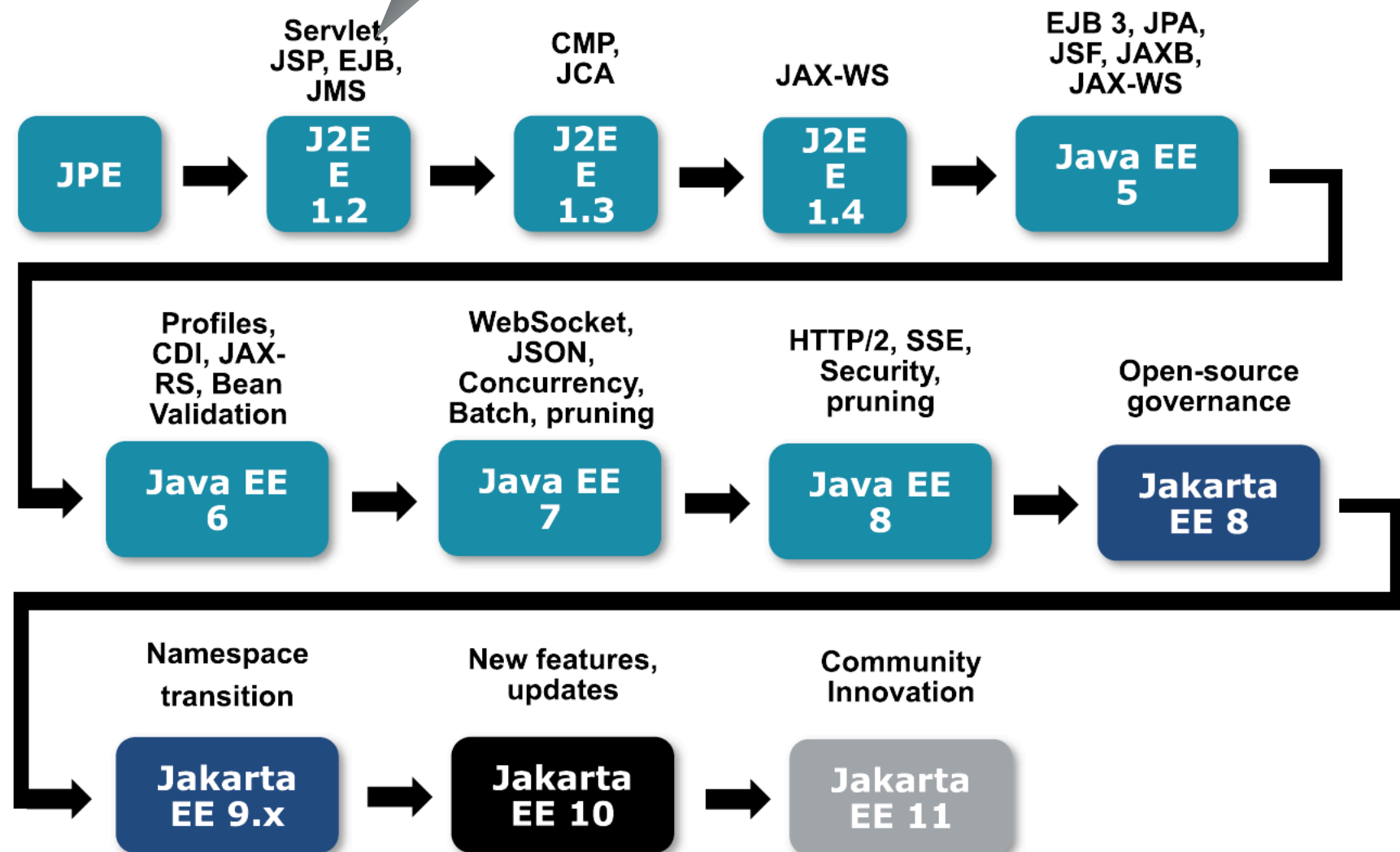
December 17, 1999

J2SE 1.2 (Play Area)

Jakarta EE Evolution



JAKARTA® EE



InfoQ



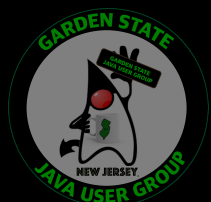
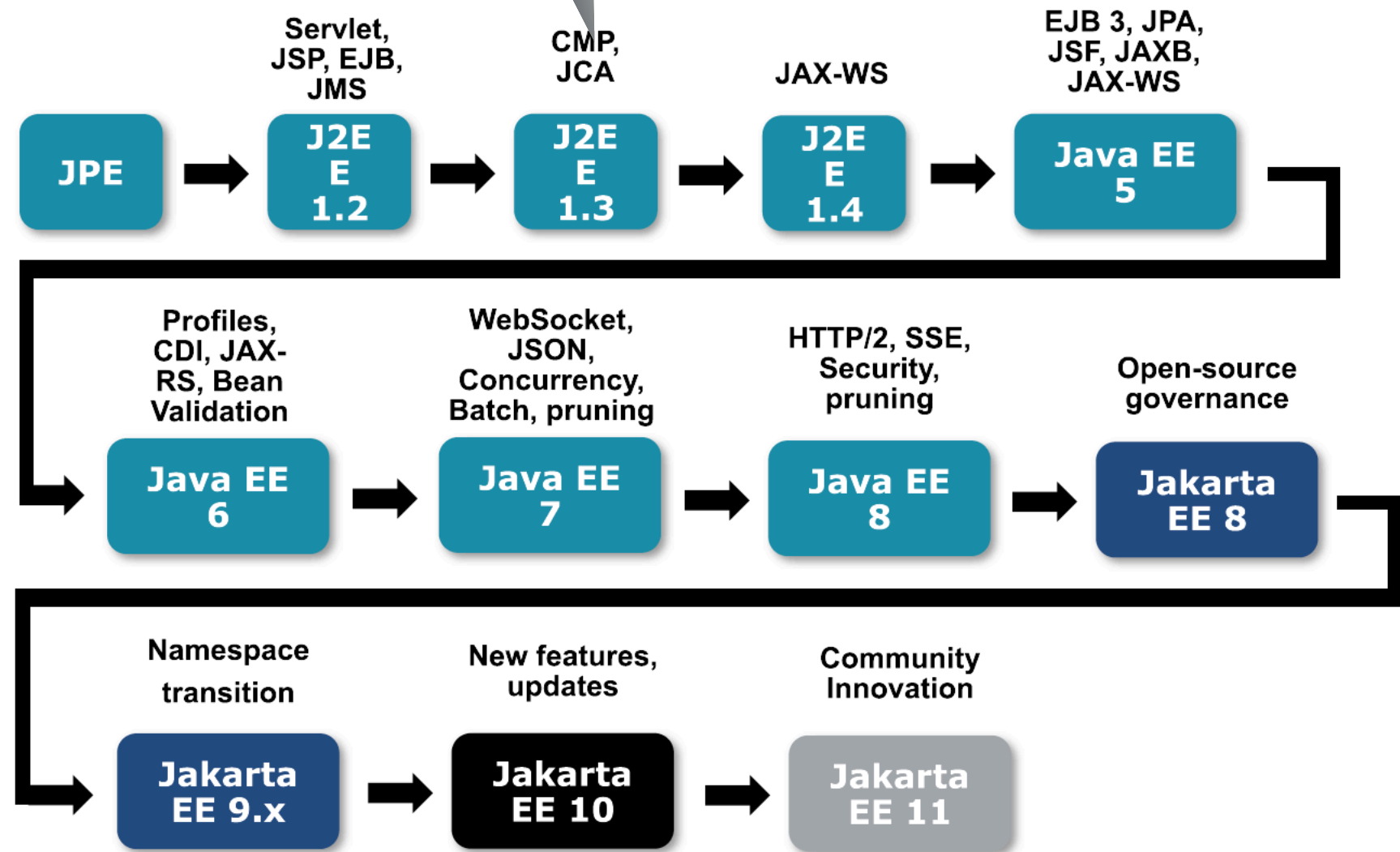
September 24, 2001

J2SE 1.3 (Kestrel)

Jakarta EE Evolution



JAKARTA EE



InfoQ



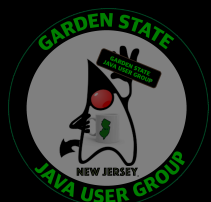
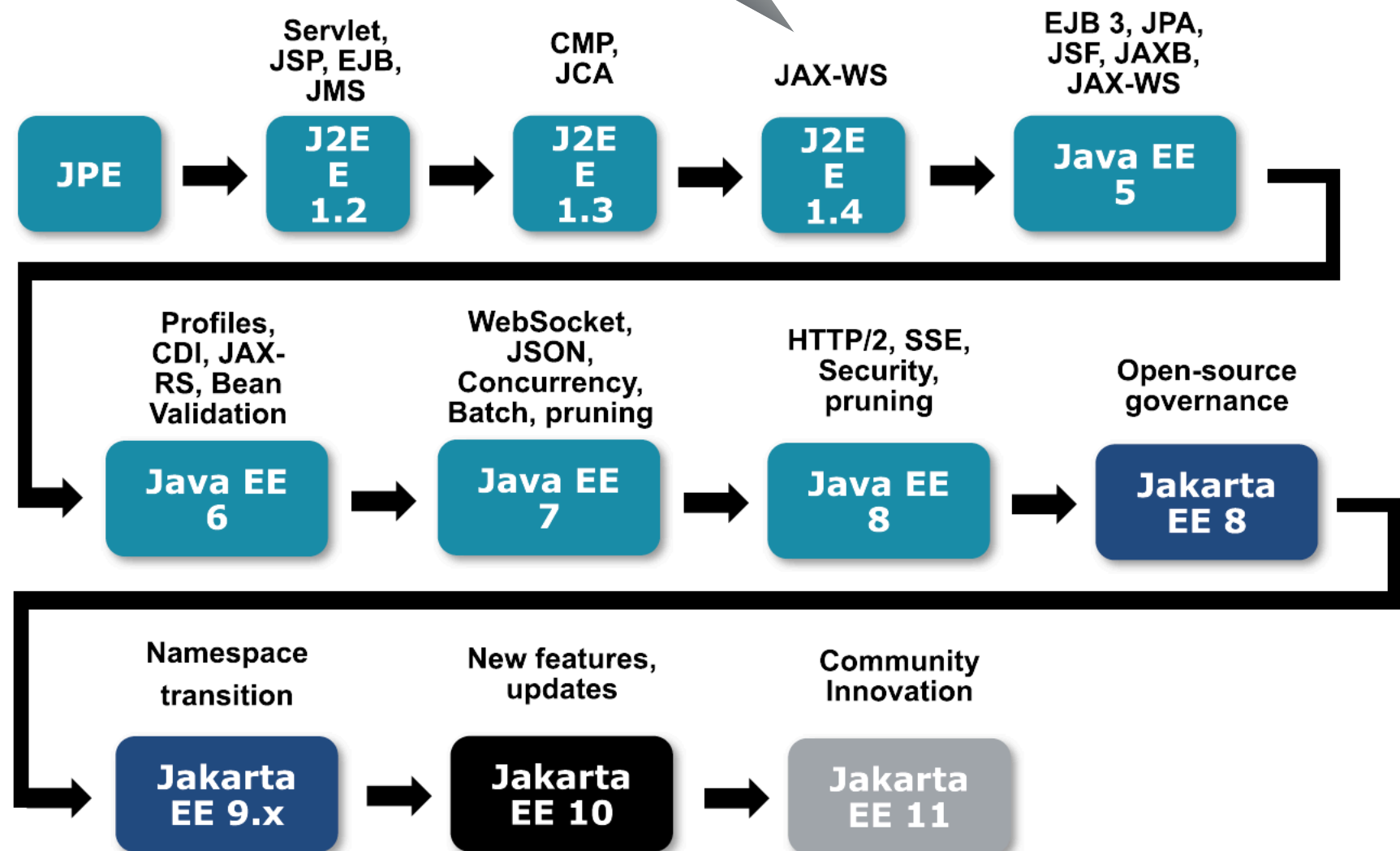
November 11, 2003

J2SE 1.4 (Merlin)

Jakarta EE Evolution



JAKARTA® EE



InfoQ



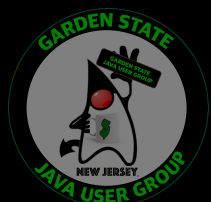
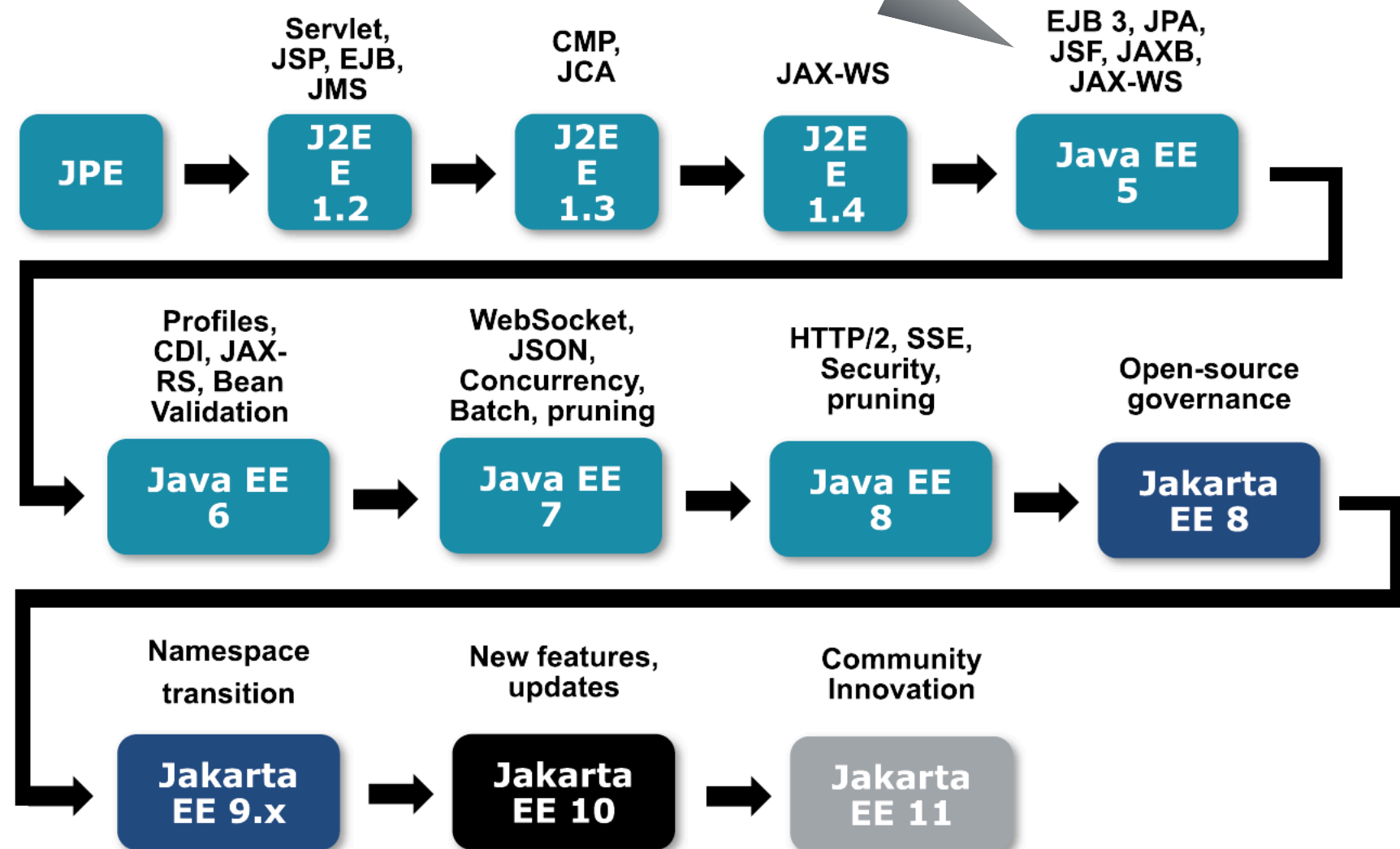
May 11, 2006

J2SE 5.0 (Tiger)

Jakarta EE Evolution



JAKARTA® EE



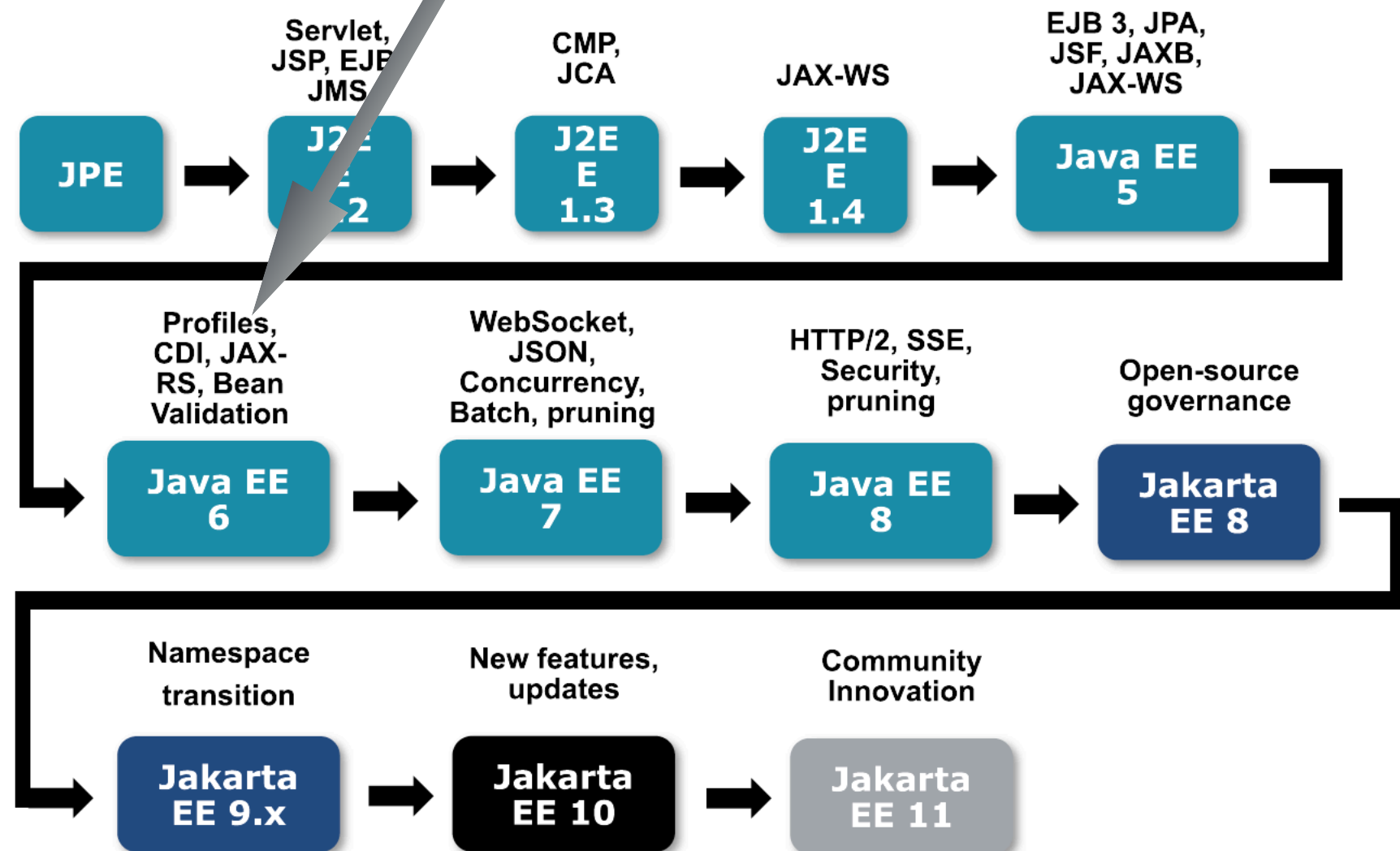
InfoQ



Jakarta EE Evolution



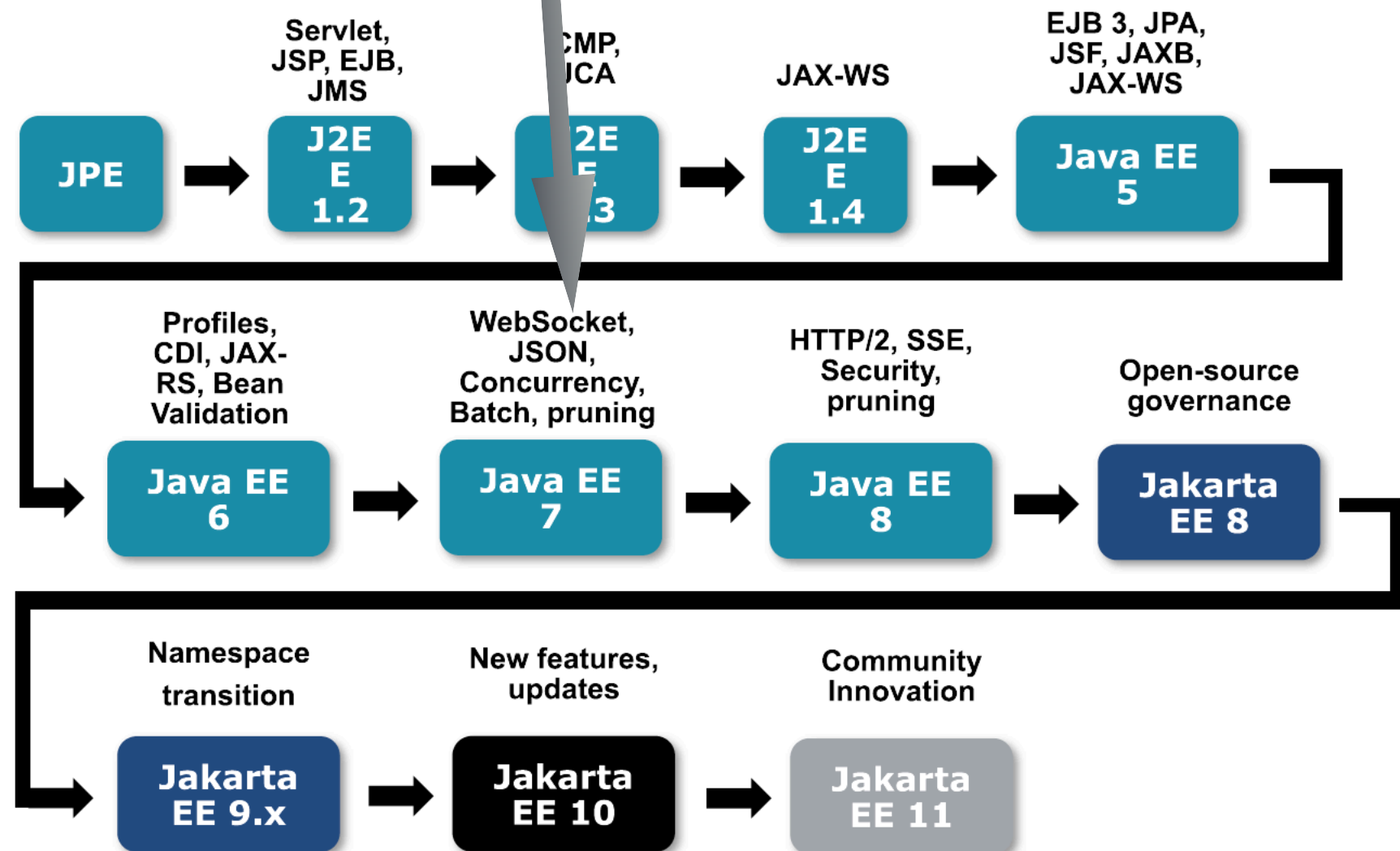
JAKARTA® EE



Jakarta EE Evolution



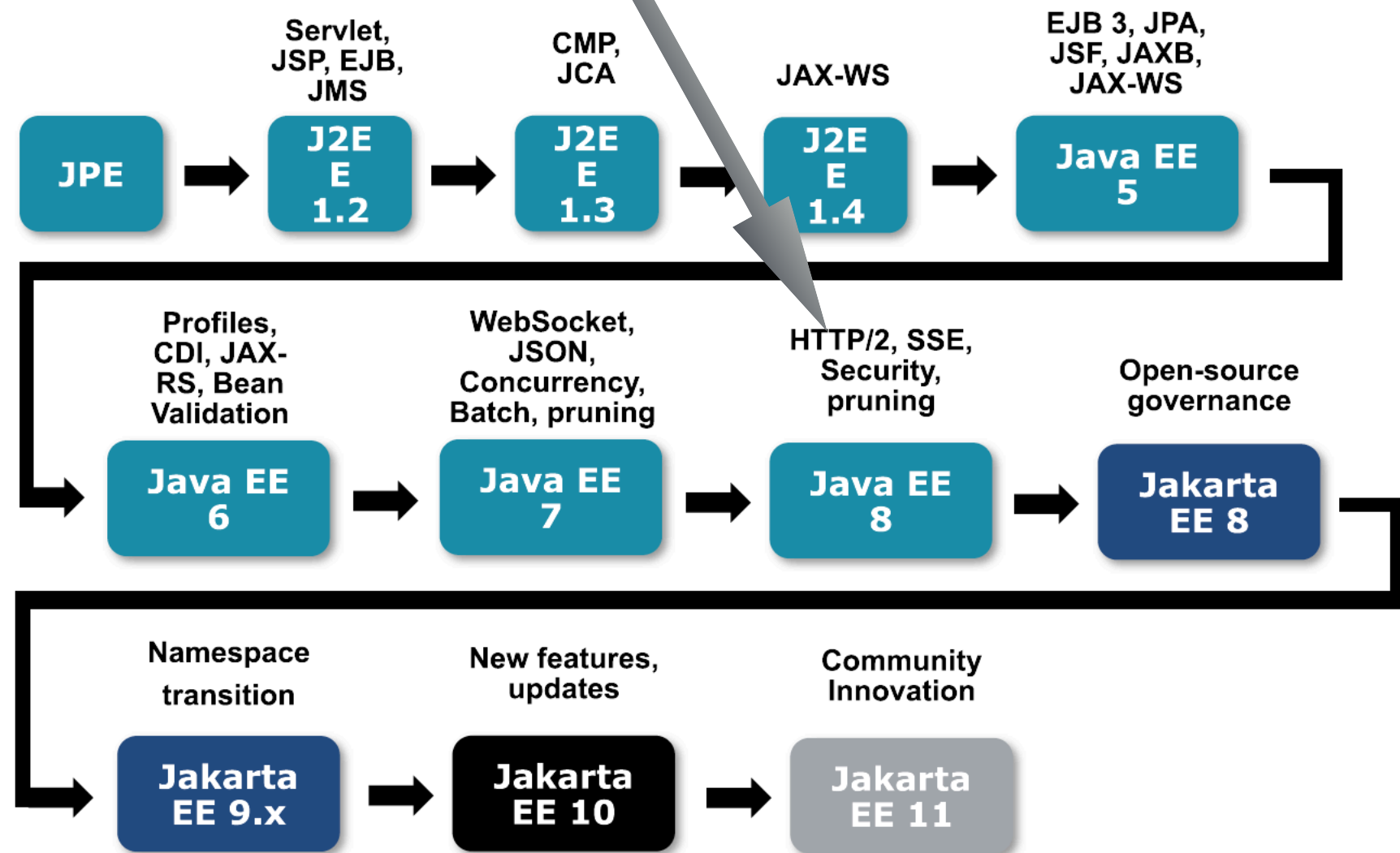
JAKARTA® EE



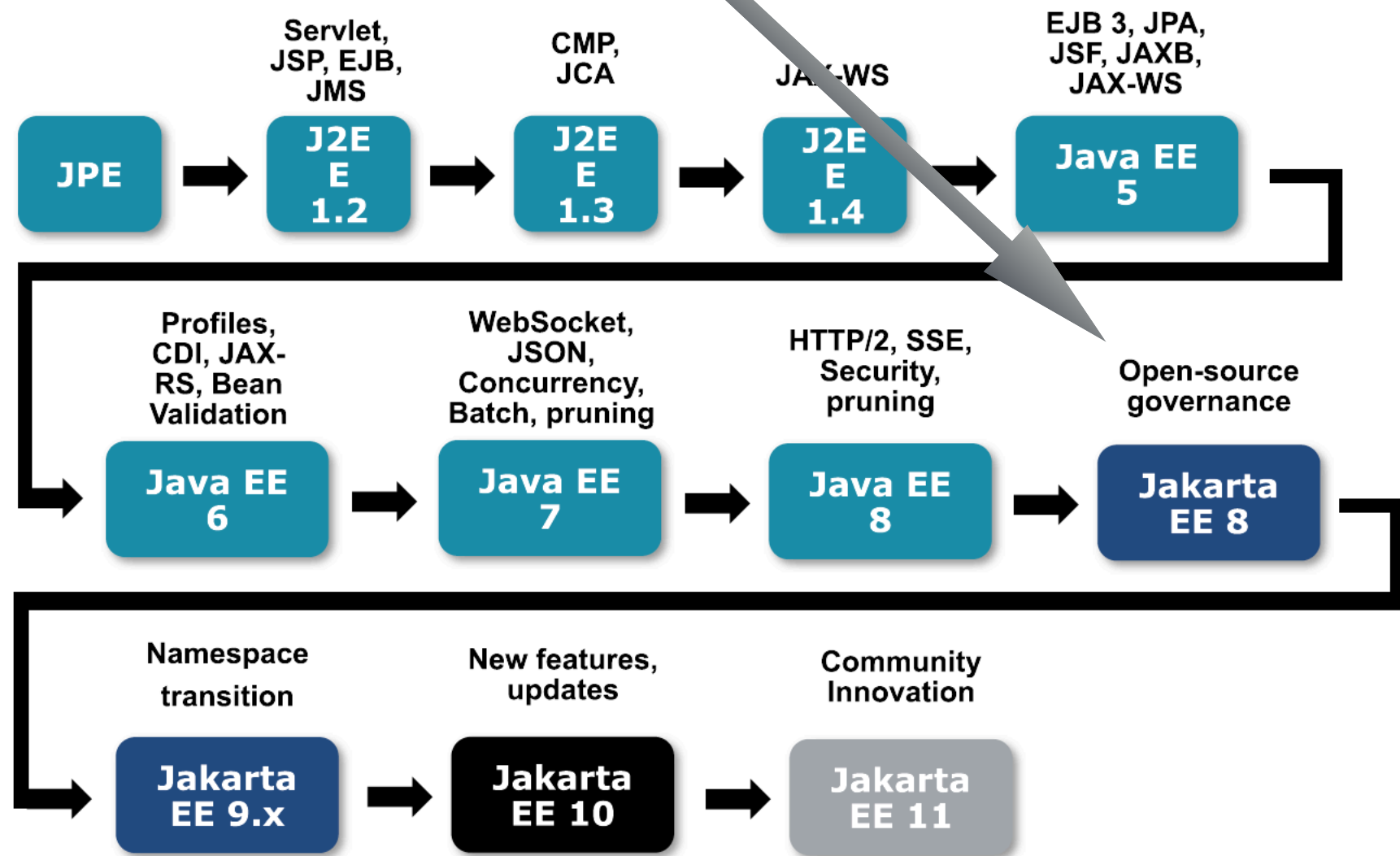
Jakarta EE Evolution



JAKARTA® EE



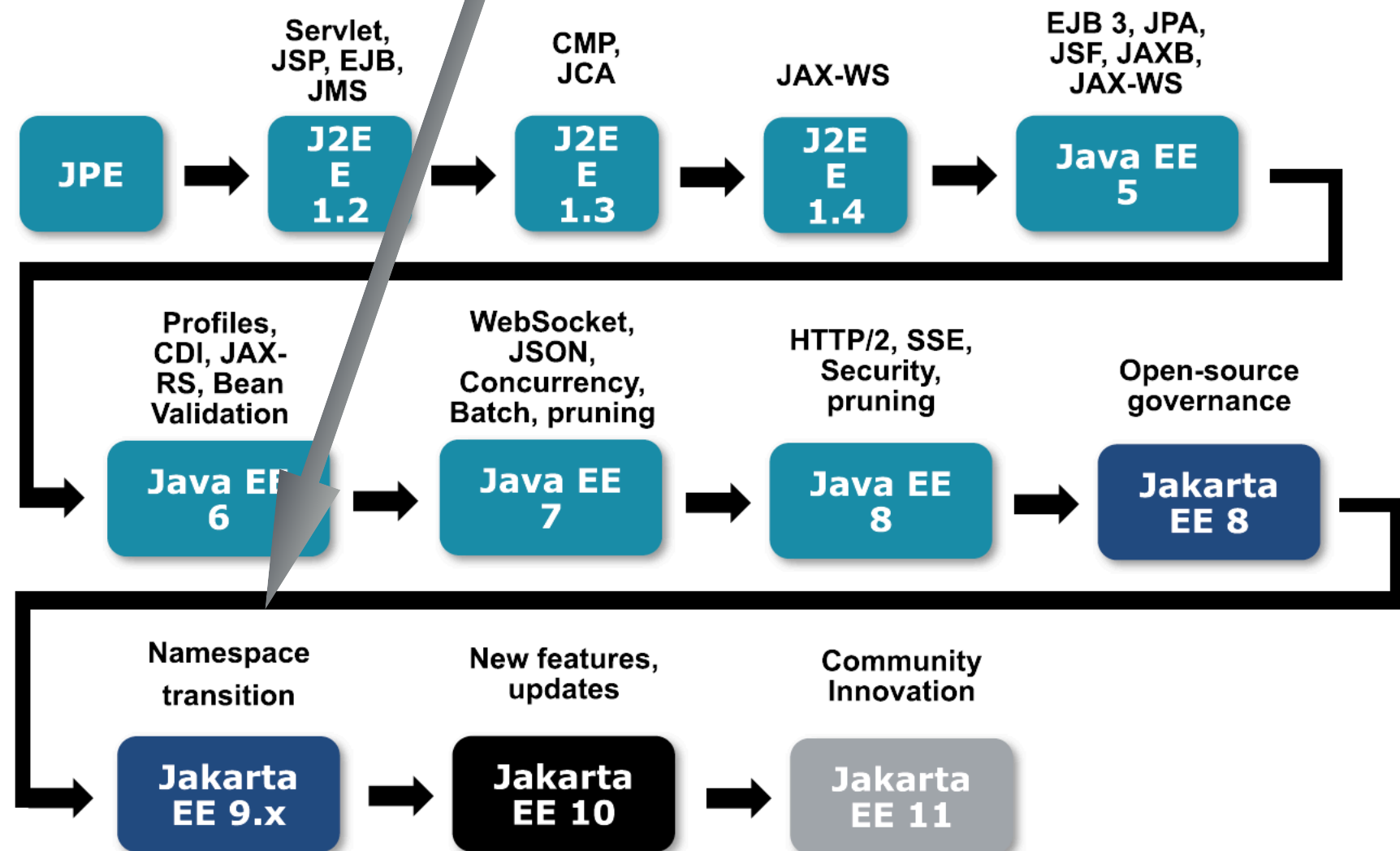
Jakarta EE Evolution



Jakarta EE Evolution



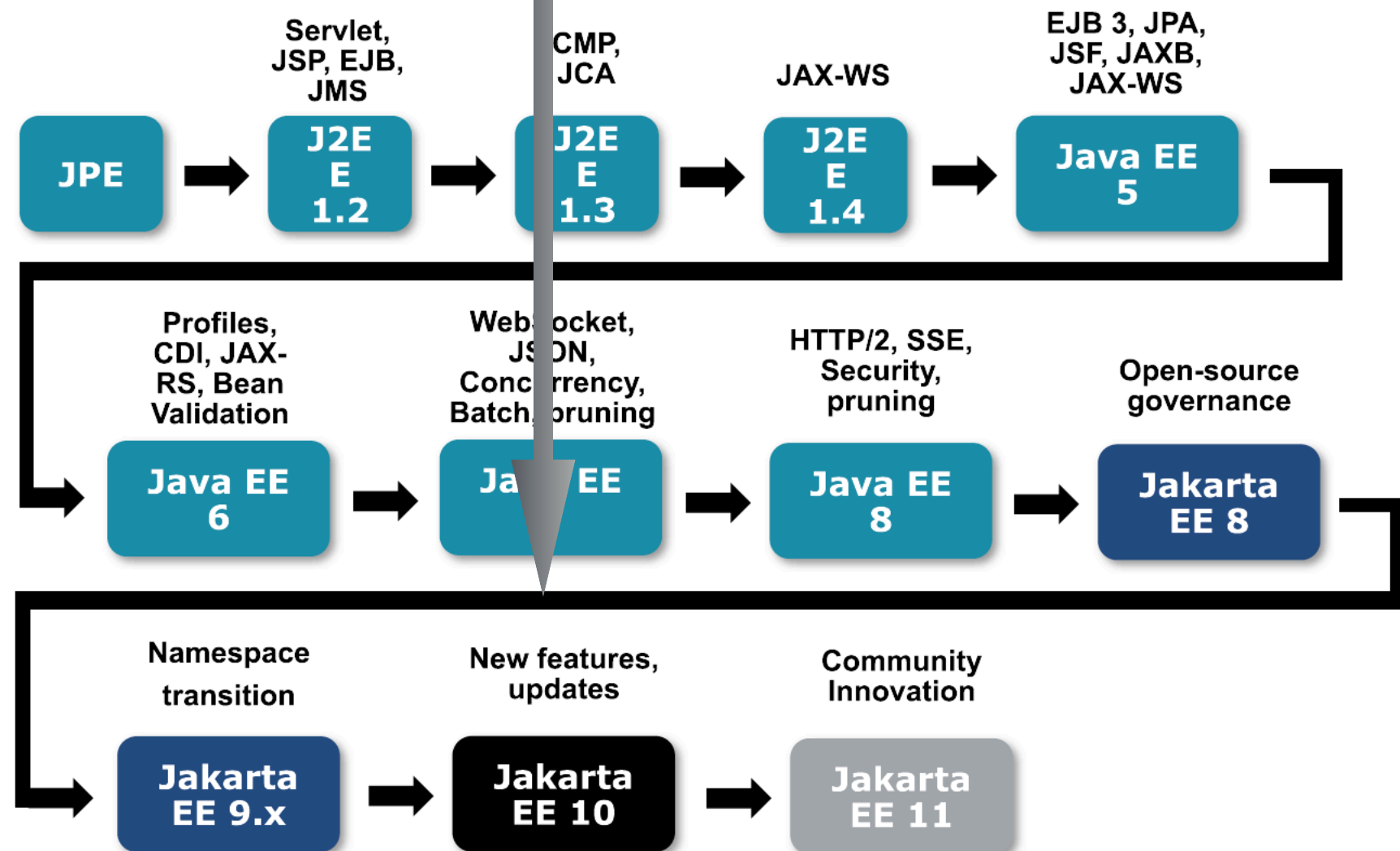
JAKARTA EE



Jakarta EE Evolution



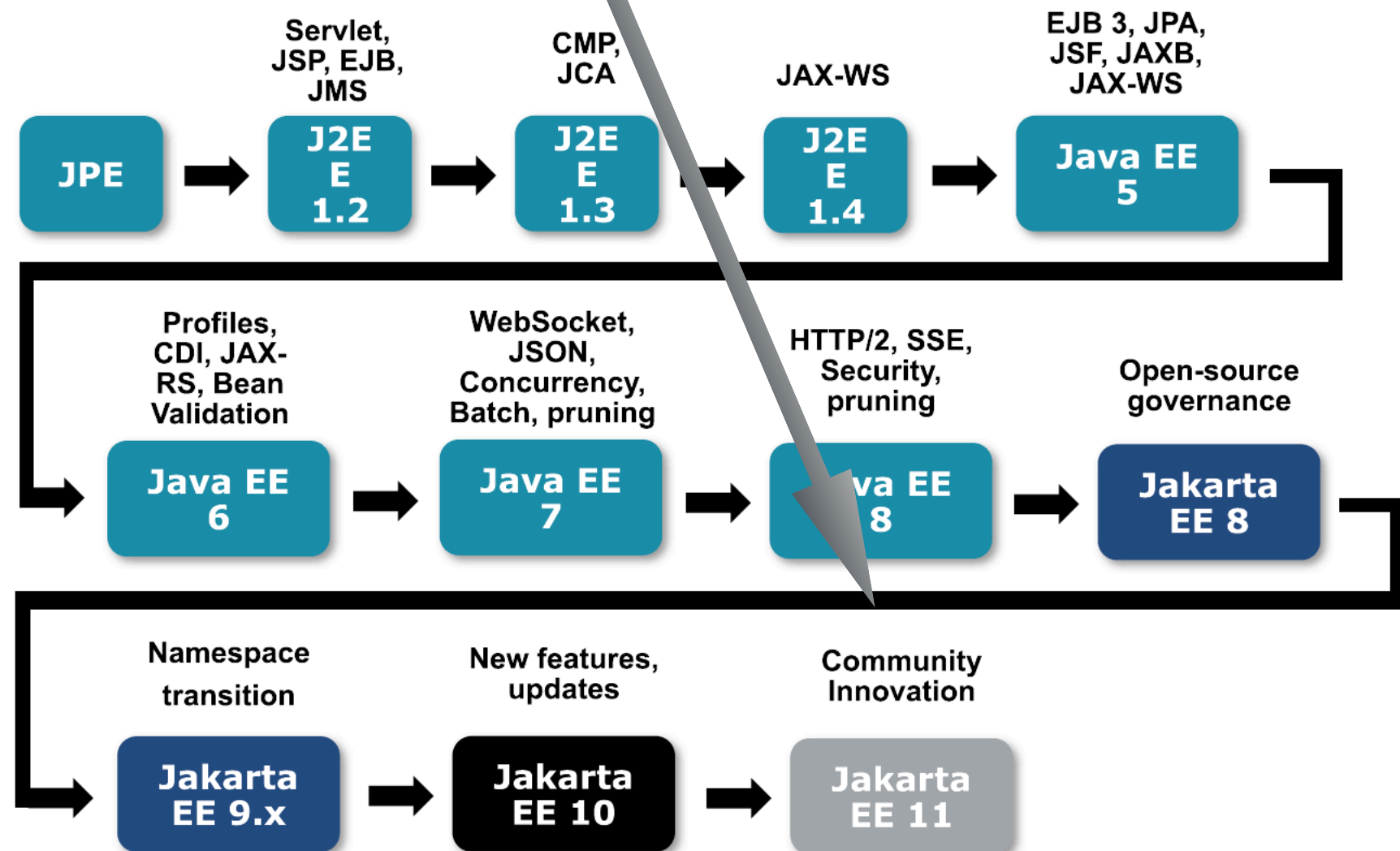
JAKARTA® EE



Jakarta EE Evolution



JAKARTA® EE



Objectives

- Evolution of Java EE/Jakarta EE 
- Jakarta EE Profiles
- What's New in Jakarta EE 11
- Updated Specifications in Jakarta EE 11
- Highlighted Specifications with Demos





Jakarta EE Profiles



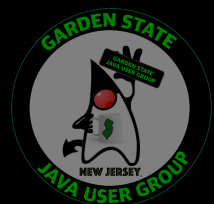
Jakarta EE Profiles

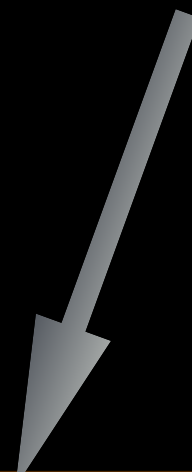
- Jakarta EE Platform
- Jakarta EE Web Profile
- Jakarta EE Core Profile



Jakarta EE Platform

- Defines a standard platform for hosting all Jakarta EE applications
- Designed for developers who require the full set of Jakarta EE specifications for developing enterprise applications





	Authentication 3.1	Data 1.0	
Authorization 3.0	Concurrency 3.1	Persistence 3.2	RESTful Web Services 4.0
Activation 2.1	CDI 4.1	Pages 4.0	JSON Processing 2.1
Batch 2.1	Expression Language 6.0	WebSocket 2.2	JSON Binding 3.0
Connectors 2.1	Faces 4.1	Validation 3.1	Annotations 3.0
Mail 2.1	Security 4.0	Debugging Support 2.0	Interceptors 2.2
Messaging 3.1	Servlet 6.1	Enterprise Beans Lite 4.0	Dependency Injection 2.0
Enterprise Beans 4.0	Standard Tag Libraries 3.0	Transactions 2.0	CDI Lite 4.1

Updated

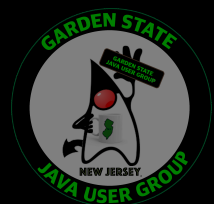
Not Updated

New



Jakarta EE Web Profile

- Defines a subset of the Jakarta EE Platform that contains web technologies specifically targeted for developing web applications

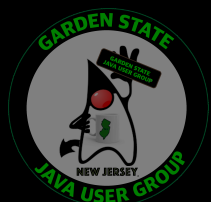


Authentication 3.1	Data 1.0	
Concurrency 3.1	Persistence 3.2	RESTful Web Services 4.0
CDI 4.1	Pages 4.0	JSON Processing 2.1
Expression Language 6.0	WebSocket 2.2	JSON Binding 3.0
Faces 4.1	Validation 3.1	Annotations 3.0
Security 4.0	Debugging Support 2.0	Interceptors 2.2
Servlet 6.1	Enterprise Beans Lite 4.0	Dependency Injection 2.0
Standard Tag Libraries 3.0	Transactions 2.0	CDI Lite 4.1

Updated

Not Updated

New

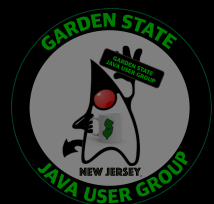


InfoQ



Jakarta EE Core Profile

- Defines a subset of the Jakarta EE Platform specifications targeting smaller runtimes suitable for microservices and ahead-of-time compilation
- Focused on providing a minimal basis for cloud native runtimes, including runtimes that support build time applications
- Introduced in Jakarta EE 10



RESTful Web Services 4.0

JSON Processing 2.1

JSON Binding 3.0

Annotations 3.0

Interceptors 2.2

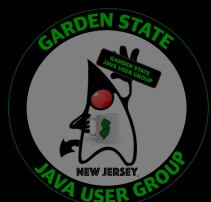
Dependency Injection 2.0

CDI Lite 4.1

Updated

Not Updated

New



InfoQ



Objectives

- Evolution of Java EE/Jakarta EE 
- Jakarta EE Profiles 
- What's New in Jakarta EE 11
- Updated Specifications in Jakarta EE 11
- Highlighted Specifications with Demos



What's New in Jakarta EE 11



InfoQ



Release Timeline (i)

- First milestone release:
 - December 2023 ✓
- Second milestone release:
 - March 2024 ✓
- Third milestone release:
 - April 2024 ✓



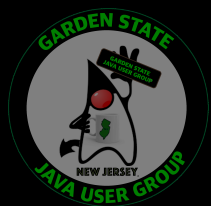
Release Timeline (ii)

- Fourth milestone release:
 - May 2024 ✓
- Final GA release:
 - June/July 2024



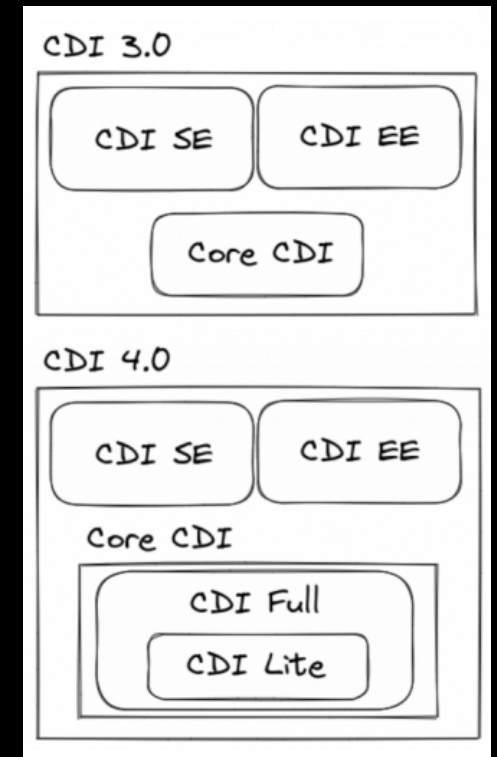
What's New (i)

- New specifications
 - Jakarta Data | MicroProfile JWT Bridge ???
- Removal of legacy `@ManagedBean` annotation
 - replace with CDI across the entire platform
- References to `SecurityManager` class have been removed



What's New (ii)

- Clarify JPMS Requirements
- CDI Lite
- Optional features
 - Jakarta SOAP with Attachments
 - Jakarta XML Web Services
 - Jakarta XML Binding



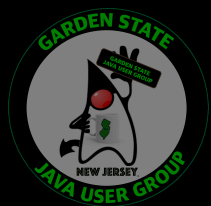
Spec Updates (i)

- Jakarta Validation
 - supports validating records
- Jakarta Persistence
 - Records will be usable as an embeddable type
- Jakarta Concurrency
 - support for Virtual Threads



Spec Updates (ii)

- Jakarta RESTful Web Services
 - improved integration with Jakarta CDI and Jakarta Concurrency
- Jakarta Security
 - APIs for the authorization theme, including interceptors and an abstraction for the permission store
 - MicroProfile JWT and Jakarta Security



Objectives

- Evolution of Java EE/Jakarta EE ✓
- Jakarta EE Profiles ✓
- What's New in Jakarta EE 11 ✓
- Updated Specifications in Jakarta EE 11
- Highlighted Specifications with Demos



Updated Specifications in Jakarta EE 11



InfoQ



Updated Specifications (i)

- Jakarta Annotations 3.0
- Jakarta Authentication 3.1
- Jakarta Authorization 3.0
- Jakarta Concurrency 3.1
- Jakarta Contexts and Dependency Injection 4.1
- Jakarta Data 1.0
- Jakarta Expression Language 6.0
- Jakarta Faces 4.1*
- Jakarta Interceptors 2.2
- Jakarta Persistence 3.2



Updated Specifications (ii)

- Jakarta RESTful Web Services 4.0
- Jakarta Security 4.0
- Jakarta Server Pages 4.0
- Jakarta Servlet 6.1
- Jakarta Validation 3.1
- Jakarta WebSocket 2.2



Objectives

- Evolution of Java EE/Jakarta EE 
- Jakarta EE Profiles 
- What's New in Jakarta EE 11 
- Updated Specifications in Jakarta EE 11 
- Highlighted Specifications with Demos



Highlighted Specifications with Demos



InfoQ





InfoQ





InfoQ



Payara Cloud

- Next generation of fully-managed cloud native application runtime
- Offers a flexible and powerful way to easily run Jakarta EE apps on the cloud without Kubernetes
- Free 15-day trial



MicroProfile JWT Bridge



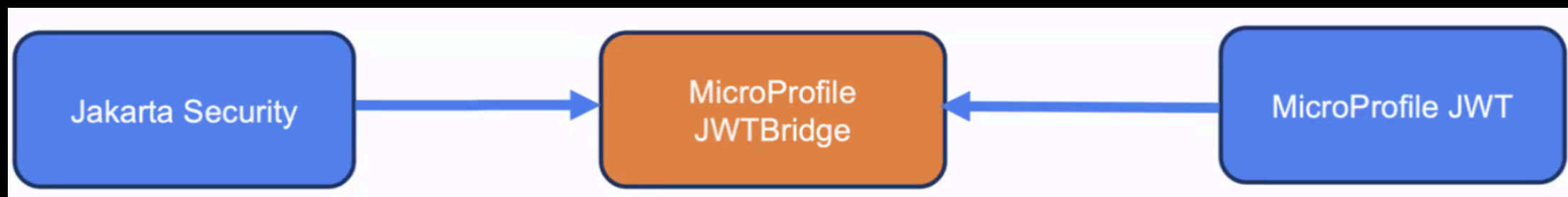
What is MicroProfile JWT Bridge? (i)

- Collaboration between Jakarta EE and MicroProfile Working Groups
- Enable Jakarta Security to build on MicroProfile JWT Authentication to provide seamless integrations
 - eliminates duplication of effort and circular dependencies

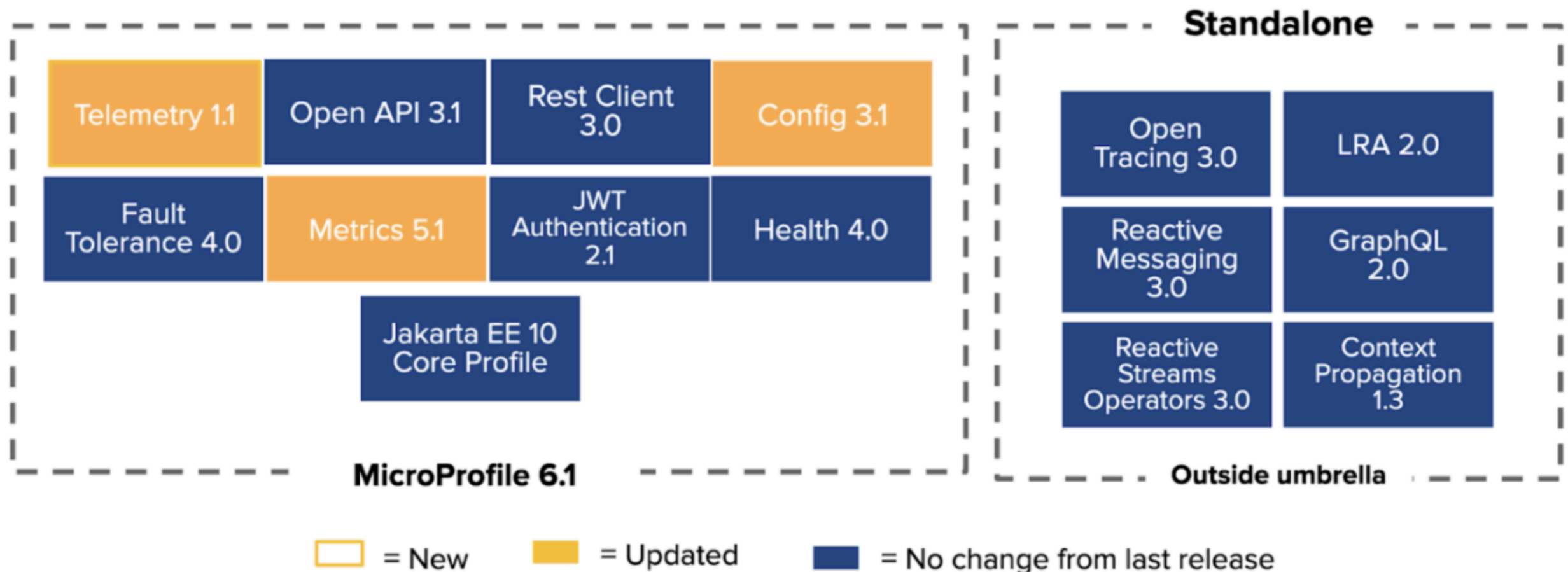


What is MicroProfile JWT Bridge? (ii)

- New standalone specification in MicroProfile
- Move the optional section of MicroProfile JWT Authentication to the new bridge specification together with TCKs



MicroProfile 6.1



Live Demo



InfoQ



Jakarta Data



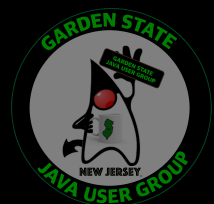
What is Jakarta Data?

- Provides an API that allows easy access to database technologies
- Can split the persistence from the model with several features, such as the ability to compose custom query methods on a **Repository** interface where the framework will implement it



Compatible Implementations

- Previous version: N/A
- Current version: 1.0.1
- Compatible implementations:
 - Open Liberty 24.0.0.6-beta and Hibernate ORM 6.6.0.Final (Jakarta EE 11)



What's New? (i)

- Support for CDI extensions
- New `BasicRepository` interface
- New `insert()` and `update()` methods in the `CrudRepository` interface
- New `@Insert`, `@Update`, `@Delete` and `@Save` annotations



What's New? (ii)

- Jakarta Data Query Language (JDQL)
 - a simple language designed to be used inside the `@Query` annotation to specify the semantics of query methods of Jakarta Data repositories
 - a subset of Jakarta Persistence Query Language (JPQL)



Live Demo



InfoQ



Jakarta Security



InfoQ



What is Jakarta Security?

- Defines a standard for creating secure Jakarta EE applications in modern application paradigms



Compatible Implementations

- Previous version: 3.0.0
- Current version: 4.0.0
- Compatible implementations:
 - Eclipse Soteria 3.0.2 (Jakarta EE 10)
 - GlassFish 8.0.0-M6 and Eclipse Soteria 4.0.0 (Jakarta EE 11)



What's New?

- Multiple authentication mechanisms
 - a new API that supports the user to choose
- Qualifiers for built-in authentication mechanisms
- References to `SecurityManager` class have been removed



Live Demo



InfoQ



Jakarta Servlet



InfoQ



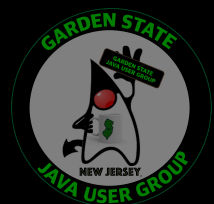
What is Jakarta Servlet?

- Defines a server-side API for handling HTTP requests and responses



Compatible Implementations

- Previous version: 6.0.0
- Current version: 6.1.0
- Compatible implementations:
 - GlassFish 7.0.0-M4 (Jakarta EE 10)
 - Tomcat 11.0.0-M20 (Jakarta EE 11)



What's New?

- Allow control of status code and response body when sending a redirect
- Constants to support new HTTP status codes
- References to `SecurityManager` class have been removed



Live Demo



InfoQ



Objectives

- Evolution of Java EE/Jakarta EE ✓
- Jakarta EE Profiles ✓
- What's New in Jakarta EE 11 ✓
- Updated Specifications in Jakarta EE 11 ✓
- Highlighted Specifications with Demos ✓



Want to Get Involved?

- Contribute to open source!



- Share
- Embrace
- Engage
- Develop

Tanja Obradovic
Jakarta EE Program Manager at
Eclipse Foundation

- Share your own Jakarta EE experiences!

Acknowledgements (i)

- Steve Millidge
 - CEO at Payara
- Ivar Grimstad
 - Jakarta EE Developer Advocate at Eclipse Foundation
- Adam Yoho
 - Back End Developer at IBM



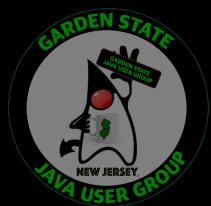
Acknowledgements (ii)

- Sergey Beryozkin
 - Principal Software Engineer at Red Hat
- Otávio Santana
 - Independent Software Consultant at OS Expert
- Maximillian Arruda
 - Senior Backend Software Engineer at Digibee



Acknowledgements (iii)

- Arjan Tijms
 - Director at OmniFishEE
- Luis Neto
 - Java Software Engineer at Payara
- David Blevins
 - CEO at Tomitribe



Contact Info

`mike@redlich.net`

`@mpredli`

`redlich.net`

`redlich.net/portfolio`

`github.com/mpredli01`



Thanks!



Backup



InfoQ



Michael Redlich

Jakarta EE 11: Going Beyond the Era of Java EE

NOV
21-23

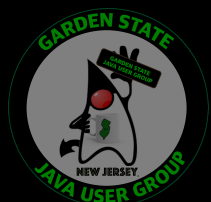


JCON 2023
WORLD
ONLINE

JAVAPRO

#JCON2023

www.jcon.one



InfoQ





VIRTUAL PAYARA CONFERENCE

Jakarta EE 11: Going Beyond the Era of Java EE

December 14 2023

3:20 PM GMT



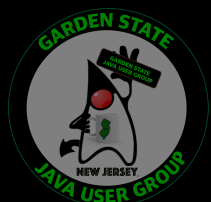
JAKARTA EE

EMPOWERING ENTERPRISE
INNOVATION WITH JAKARTA EE



**MICHAEL
REDLICH**

Java Champion



InfoQ



Jakarta Annotations



What is Jakarta Annotations?

- Defines a collection of annotations representing common semantic concepts that enable a declarative style of programming that applies across a variety of Java technologies



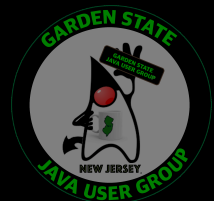
Compatible Implementations

- Previous version: 2.1.0
- Current version: 3.0.0
- Compatible implementations:
 - GlassFish 7.0.0-M1 (Jakarta EE 10)
 - GlassFish 8.0.0-M3 (Jakarta EE 11)



What's New?

- Removal of the deprecated `@ManagedBeans` annotation



Jakarta Authentication



What is Jakarta Authentication?

- Defines a general low-level SPI for authentication modules:
 - controllers that interact with a caller and the environment of a container to acquire the caller's credentials, validate them, and pass an authenticated identity to the container



Compatible Implementations

- Previous version: 3.0.0
- Current version: 3.1.0
- Compatible implementations:
 - GlassFish 7.0.0-M7 (Jakarta EE 10)
 - GlassFish 8.0.0-M6 and Eclipse Epicyro 3.1.0 (Jakarta EE 11)



What's New?

- Minor updates to support the overall goals of Jakarta Security
- Clarifications around state and concurrency
- References to `SecurityManager` class have been removed

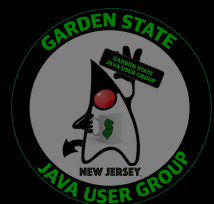


Jakarta Authorization



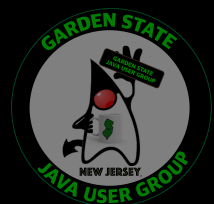
What is Jakarta Authorization?

- Defines a general low-level SPI for authorization modules:
 - repositories of permissions facilitating subject-based security by determining whether a given subject has a defined permission
 - algorithms to transform security constraints for specific containers (such as Jakarta Servlet or Jakarta Enterprise Beans) into these permissions



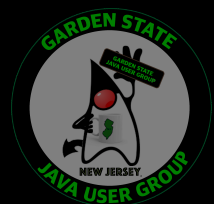
Compatible Implementations

- Previous version: 2.1.0
- Current version: 3.0.0
- Compatible implementations:
 - GlassFish 7.0.0-M4 and Eclipse Exousia 2.1.0-M1 (Jakarta EE 10)
 - GlassFish 8.0.0-M5 and Eclipse Exousia 3.0.0-M3 (Jakarta EE 11)



What's New? (i)

- Future-proof the specification and make it more suitable for cloud deployments
 - achieved by adding an API for programmatically registering policy providers to create a replacement for the deprecated `Policy` class
- Standardize the context ID for Servlet containers



What's New? (ii)

- Several new convenience methods to make the API easier to use
- References to `SecurityManager` class have been removed



Jakarta Concurrency



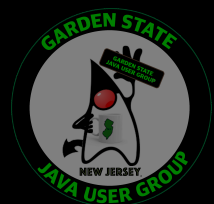
What is Jakarta Concurrency?

- Provides a mechanism for using concurrency from application components without compromising container integrity while still preserving the fundamental benefits of the Jakarta EE platform



Compatible Implementations

- Previous version: 3.0.0
- Current version: 3.1.0
- Compatible implementations:
 - Open Liberty 22.0.0.5-beta (Jakarta EE 10)
 - Open Liberty 24.0.0.6-beta (Jakarta EE 11)



What's New? (i)

- Integration with Java 21 Virtual Threads
- Support for the Java Flow class and the MicroProfile Reactive Streams Operators and MicroProfile Context Propagation specifications



What's New? (ii)

- Replace more features from EJB
 - such as the `@Schedule` annotation
- Become more CDI-centric
 - by adding the ability to inject default resource implementations of the `ManagedExecutorService` and `ContextService` interfaces



Jakarta Contexts and Dependency Injection (CDI)



What is Jakarta CDI?

- Specifies a means for obtaining objects in such a way to maximize reusability, testability and maintainability as compared to traditional approaches such as constructors, factories, and service locators like the Java Naming and Directory Interface (JNDI)



Compatible Implementations

- Previous version: 4.0.0
- Current version: 4.1.0
- Compatible implementations:
 - Weld 5.0.0.SP2 (Jakarta EE 10)
 - Weld 6.0.0.Beta1 (Jakarta EE 11)



What's New?

- Add methods to **BeanConfigurator** interface for applying decorators
- Support for **@Priority** annotation on producers



Jakarta Expression Language (EL)



What is Jakarta EL (i)?

- Defines a simple language to meet the needs of the presentation layer in web applications that features:
 - a simple syntax restricted to the evaluation of expressions
 - variables and nested properties



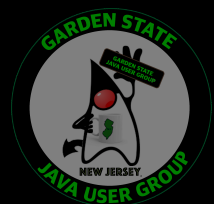
What is Jakarta EL (ii)?

- Defines a simple language to meet the needs of the presentation layer in web applications that features:
 - relational, logical, arithmetic, conditional and empty operators
 - functions implemented as static methods on Java classes



Compatible Implementations

- Previous version: 5.0.0
- Current version: 6.0.0
- Compatible implementations:
 - Eclipse ExpressLy 5.0.0-M2 (Jakarta EE 10)
 - Apache Tomcat 11.0.0-M18 (Jakarta EE 11)



What's New?

- The `java.desktop` module is no longer required at runtime
- A new property, `length`, is now supported for arrays



Jakarta Faces



What is Jakarta Faces?

- Defines a Model-View-Controller (MVC) framework for building user interfaces for web applications that include UI components, state management, event handling, input validation, page navigation, and support for internationalization and accessibility
- Formerly known as Jakarta Server Faces



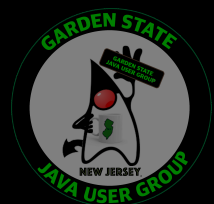
Compatible Implementations

- Previous version: 4.0.0
- Current version: 4.1.0
- Compatible implementations:
 - Eclipse Mojarra 4.0.4 (Jakarta EE 10)
 - GlassFish 8.0.0-M6 and Eclipse Mojarra 4.1.0 (Jakarta EE 11)



What's New?

- Require firing events for Jakarta CDI `@Initialized`, `@BeforeDestroyed` and `@Destroyed` for build-in scopes
- Deprecate the unused `PreDestroyCustomScopeEvent` and `PostConstructCustomScopeEvent` classes



Jakarta Interceptors



What is Jakarta Interceptors?

- Defines a means of interposing on business method invocations and specific events, such as lifecycle and timeout events, that occur on instances of Jakarta EE components and other managed classes



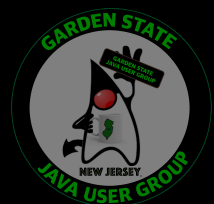
Compatible Implementations

- Previous version: 2.1.0
- Current version: 2.2.0
- Compatible implementations:
 - Weld 5.0.0.Beta1 (Jakarta EE 10)
 - Weld 6.0.0.Beta1 (Jakarta EE 11)

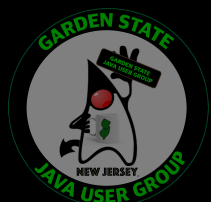


What's New?

- Updated dependencies to align with Jakarta EE 11
- New standard overloaded `getInterceptorBindings()` accessor methods to the `InvocationContext` interface for interceptor bindings



Jakarta Pages

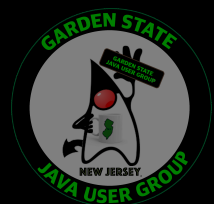


InfoQ



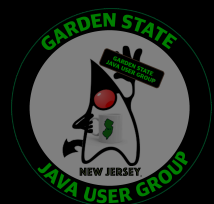
What is Jakarta Pages?

- Defines a template engine for web applications that supports mixing of textual content (including HTML and XML) with custom tags, expression language, and embedded Java code, that is ultimately compiled into a Jakarta Servlet
- Formerly known as Jakarta Server Pages



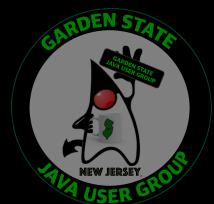
Compatible Implementations

- Previous version: 3.1.0
- Current version: 4.0.0
- Compatible implementations:
 - Eclipse WaSP 3.1.0-M3 (Jakarta EE 10)
 - Apache Tomcat 11.0.0-M20 (Jakarta EE 11)



What's New?

- An updated `ErrorData` class to support the new Jakarta Servlet constant field values:
 - `ERROR_QUERY_STRING` and `ERROR_METHOD`
- All code deprecated as of Jakarta Server Pages 3.1 has been removed



Jakarta Persistence



What is Jakarta Persistence?

- Defines a standard for the management of persistence and object/relational mapping in a Java environment



Compatible Implementations

- Previous version: 3.1.0
- Current version 3.2.0
- Compatible implementations:
 - EclipseLink 4.0.0-M3 and Hibernate ORM 6.0.0.Final (Jakarta EE 10)
 - EclipseLink 5.0.0-B02 and Hibernate ORM 7.0.0.Alpha 2 (Jakarta EE 11)



What's New? (i)

- Support for Java record types as embeddable classes
- Numerous new methods added to classes/interfaces such as:
 - `CriteriaQuery`, `EntityManager`, `PersistenceUnitUtil` and `Metamodel`



What's New? (ii)

- A migration to the `java.time` API
 - deprecate the use of the Java Calendar, `java.util.Date`, `java.sql.Date`, `Time` and `Timestamp` classes
- Remove all deprecated code from Jakarta Servlet 3.1



Jakarta RESTful Web Services (JAX-RS)



What is Jakarta JAX-RS?

- Provides a foundational API to develop web services following the Representational State Transfer (REST) architectural pattern



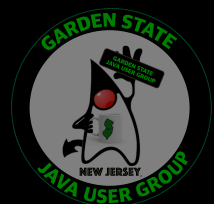
Compatible Implementations

- Previous version: 3.1.0
- Current version: 4.0.0
- Compatible implementations:
 - Eclipse Jersey 3.1.0 (Jakarta EE 10)
 - Eclipse Jersey 4.0.0-M1 and RESTEasy 7.0.0.Alpha1 (Jakarta EE 11)



What's New? (i)

- A new convenience method, `containsHeaderString()`, added to the `ContainerResponseContext` interface, for checking header value lists
- A new method, `getMatchedResourceTemplate()`, added to the `UriInfo` interface, to obtain the template for the current URI request



What's New? (ii)

- Additional TCK tests for the **ExceptionHandler** interface and the Multipart/Form-Data API
- The deprecated **@ManagedBean** annotation and dependency on the Jakarta XML Binding specification were removed

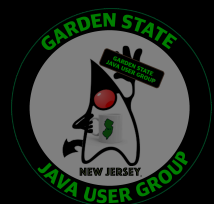


Jakarta Validation



What is Jakarta Validation? (i)

- Provides an object level constraint declaration and validation facility as well as a constraint metadata repository and query API
- Offers method and constructor validation facilities to ensure constraints on their parameters and return values



What is Jakarta Validation? (ii)

- Formerly known as Jakarta Bean Validation



Compatible Implementations

- Previous version: 3.0.0
- Current version: 3.1.0
- Compatible implementations:
 - Hibernate Validator 7.0.0.Alpha5 (Jakarta EE 9)
 - Hibernate Validator 8.0.1.Final (Jakarta EE 11)



What's New?

- Support for Java Records
- Update dependencies for Jakarta EE 11



Jakarta WebSocket



What is Jakarta WebSocket?

- Defines an API for client and server endpoints for the WebSocket protocol as defined by the Internet Engineering Task Force (IETF) RFC 6455



Compatible Implementations

- Previous version: 2.1.0
- Current version 2.2.0
- Compatible implementations:
 - Eclipse Tyrus 2.1.0-M3 (Jakarta EE 10)
 - Eclipse Tyrus 2.2.0-M1 and Apache Tomcat 11.0.0-M20 (Jakarta EE 11)



What's New? (i)

- Clarify the responsibilities for sending ping and pong messages
- Add a `getSession()` method to the `SendResult` class that returns the websocket `Session` in which the session was sent



What's New? (i)

- Clarify the behavior of the `maxMessageSize` attribute, defined in the `@OnMessage` annotation, is set to a value larger than `Integer.MAX_VALUE`
- References to `SecurityManager` class have been removed



Jakarta EE 11

Going Beyond the Era of Java EE



Michael P. Redlich