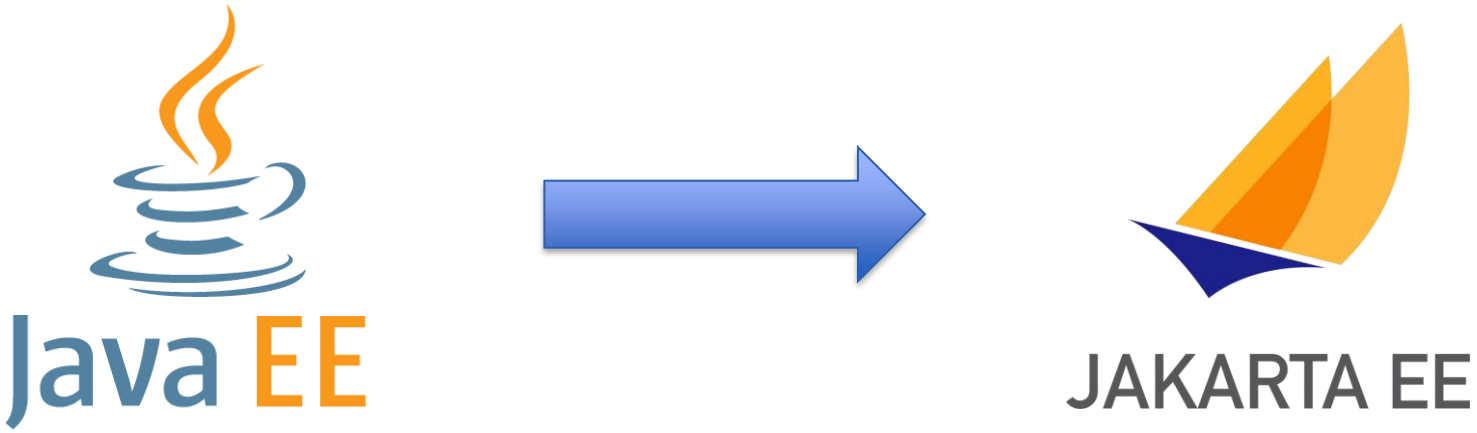
Contributor's Guide to the Jakarta EE 12 Galaxy

Reza Rahman
Jakarta EE Ambassador, Author, Blogger, Speaker
@reza_rahman, reza_rahman@mail.com

Jakarta EE

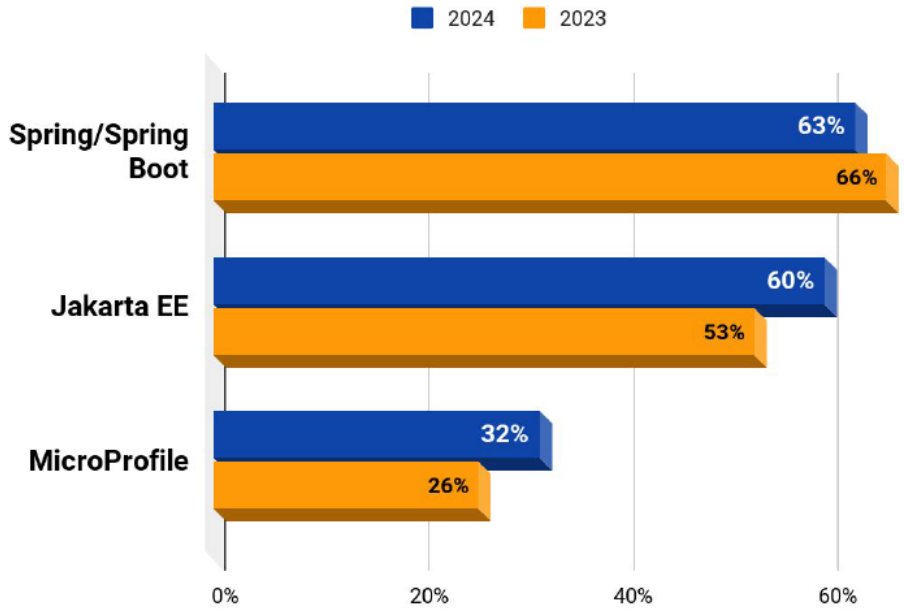
- Java EE transitioned from JCP to Eclipse Foundation as Jakarta EE
- Open governance, open source, open compatibility testing
- Well-defined specification process, clear IP flow, vendor-neutral open collaboration, level playing field
- Key stakeholders maintained if not expanded including Oracle, IBM, Payara, Tomitribe, Red Hat, Microsoft, VMware, Alibaba, London Java Community, OmniFish, SouJava, Apache
- Community participation and contribution key



https://jakarta.ee

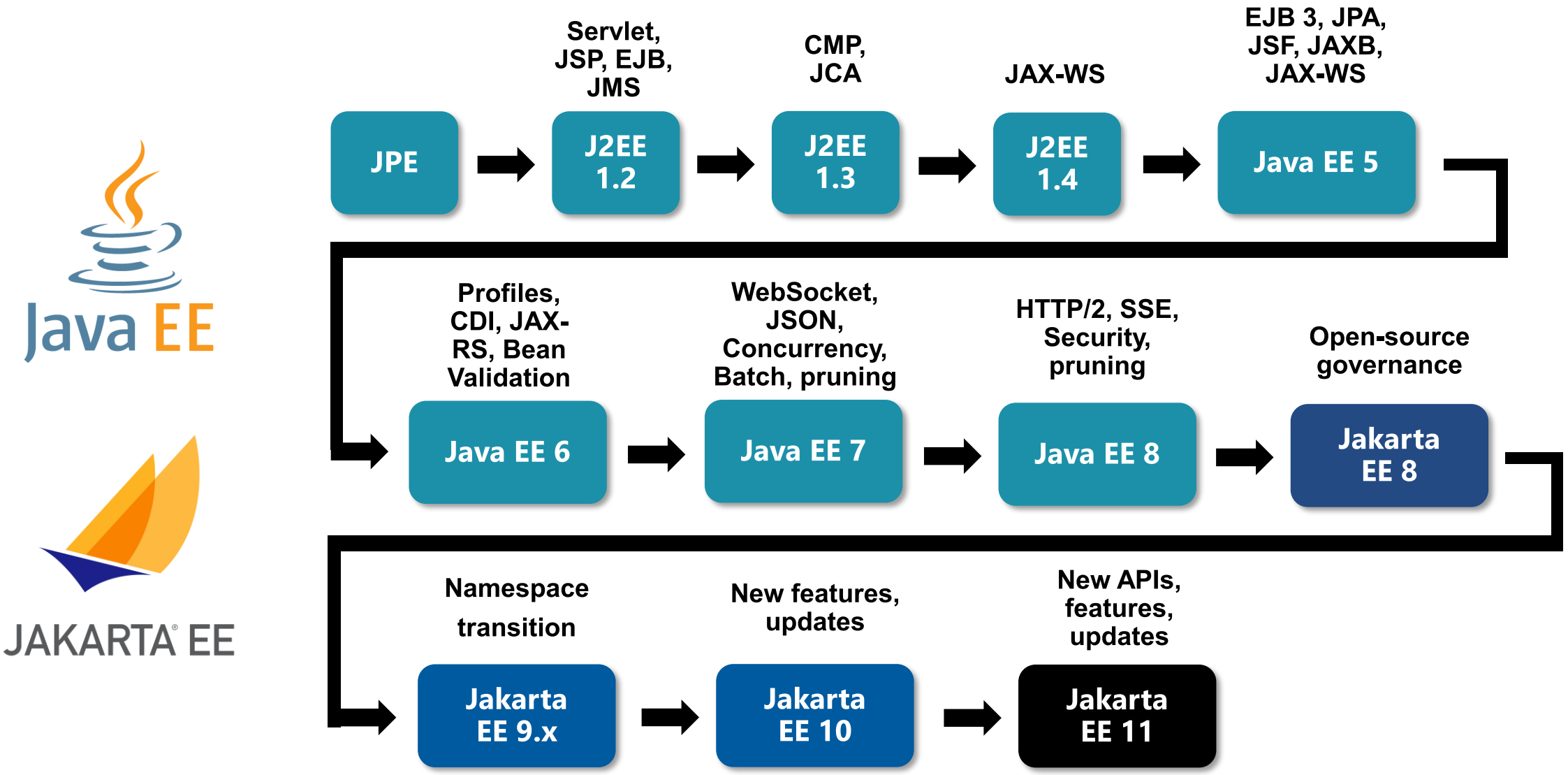
The Importance of Jakarta EE

- Jakarta EE is an important part of the Java ecosystem
- 25-35% of Java applications run on Jakarta EE runtimes
  - WildFly, Payara, GlassFish, JBoss EAP, WebSphere/Liberty, WebLogic
- 70-80% of Java applications depend on at least one or more Jakarta EE APIs
  - Tomcat, Hibernate, ActiveMQ, Jetty, CXF, Jersey, RESTEasy, Quarkus, MicroProfile, Spring Boot



2024 Jakarta EE Developer Survey: https://outreach.eclipse.foundation/jakarta-ee-developer-survey-2024

Jakarta EE Evolution



A Lively Ecosystem

Ambassadors' Jakarta EE 11 Contribution Guide

https://jakartaee-ambassadors.io/guide-to-contributing-to-jakarta-ee-11

@jee_ambassadors

Jakarta EE 10 in Context

- CDI Alignment
  - `@Asynchronous`, `@Schedule`, `@Lock`, `@MaxConcurrency` in Concurrency, `@MessageListener` in Messaging, `@RolesAllowed`, `@RunAs` in Security
  - Better CDI support in Batch, REST
- Java SE Alignment
  - `CompletionStage` in Concurrency
  - Bootstrap APIs for REST, Messaging
- Closing standardization gaps
  - OpenID Connect, JWT, in-memory identity store, batch job definition Java API, `@ManagedExecutorDefinition`, more SQL support, `multipart/form-data`
  - Core/Microservices Profile
- Deprecation/removal
  - EJB Entity Beans, embeddable EJB container, deprecated Servlet/Faces/CDI features
- Innovation
  - Repositories, NoSQL, MVC, Configuration, gRPC

Made it! **On the way** **Gap**

Jakarta EE 11 in Context

- CDI Alignment
  - `@Schedule`, `@Lock`, `@MaxConcurrency` in Concurrency, `@MessageListener` in Messaging, `@RolesAllowed`, `@RunAs` in Security
  - Better CDI support in REST, Persistence, Concurrency
- Java SE Alignment
  - Adapting to Records, Virtual Threads
  - Bootstrap API for Messaging
  - Modularity, standalone/modernized TCKs
- Closing standardization gaps
  - In-memory identity store, JWT, batch job definition Java API, `@Service`
- Deprecation/removal
  - JAX-WS (SOAP), JAXB (XML), CORBA, @ManagedBean, EJB
- Innovation
  - Repositories, NoSQL, MVC, Configuration, gRPC

Made it! **On the way** **Gap**

Jakarta EE 12 Possibilities

- CDI Alignment
  - `@Lock`, `@MaxConcurrency` in Concurrency, `@MessageListener` in Messaging, `@RolesAllowed`, `@RunAs` in Security
  - Better CDI support in REST, Concurrency
- Java SE Alignment
  - Adapting to Records in JSON Binding
  - Bootstrap API for Jakarta EE, bootstrap API for Messaging
  - Modularity, standalone/modernized TCKs
- Closing standardization gaps
  - JWT, batch job definition Java API, `@Service`
- Deprecation/removal
  - Application Client Container, EJB
- Innovation
  - Configuration, NoSQL, MVC, gRPC

Jakarta Concurrency

- CDI based, modernized equivalent for EJB `@Lock`
- Adding `@MaxConcurrency`
- CDI context propagation

CDI Based @Lock

```
@ApplicationScoped @Lock(READ)
public class CountryInfoService {
    ...
    public List<String> getStates(String country) {
        return countryStatesMap.get(country);
    }
    ...
    @Lock(WRITE)
    public void setStates(String country, List<String> states) {
        countryStatesMap.put(country, states);
    }
}
```

Jakarta Messaging

- CDI based, modernized `@MessageListener`
- Jakarta Messaging Lite for cloud native use cases
- Standalone bootstrap API
- AMQP interoperability

@MessageListener Example

```
@ApplicationScoped
@MaxConcurrency(10)
public class HandlingEventRegistrationAttemptConsumer {
    @MessageListener(
        destinationLookup="jms/HandlingEventRegistrationAttemptQueue",
        selector="source = 'mobile'",
        batchSize=10, orderBy=TIMESTAMP,
        retry=5, retryDelay=7000, deadLetterQueue="jms/DeadLetterQueue")
    public void onEventRegistrationAttempt(
        HandlingEventRegistrationAttempt... attempts) {
        ...
    }
}
```

Bootstrap API for Jakarta Messaging

```
JmsContainer container = JmsContainerFactory.newInstance();
...
JMSContext jmsContext = container.getJmsContext();
Destination handlingEventQueue = container.getDestination(
    “jms/HandlingEventRegistrationAttemptQueue” true);
...
jmsContext.createProducer()
    .setDisableMessageID(true)
    .setDisableMessageTimestamp(true)
    .setStringProperty("source", source)
    .send(handlingEventQueue, attempt);
...
container.close();
```

Jakarta Security

- JWT alignment
  - MicroProfile bridge specification
- CDI based, modernized equivalent for `@RolesAllowed`, `@RunAs`
- EL-enabled authorization annotation

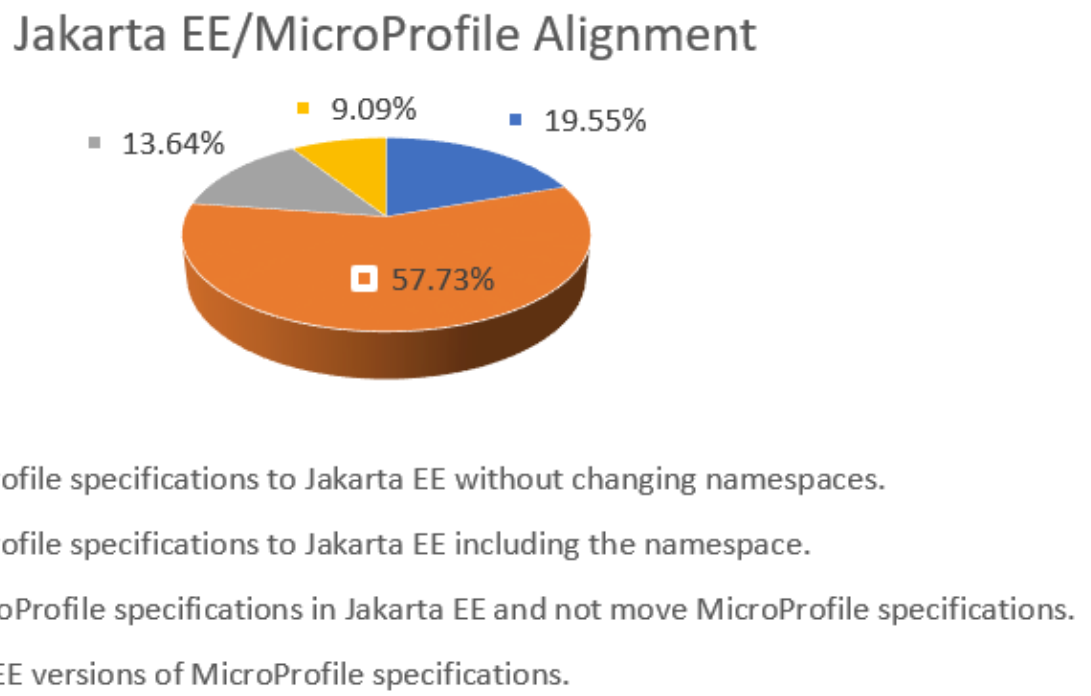
EL Enabled Security Annotations

```
@Authorized("hasRoles('Manager') && schedule.officeHours")
public void transferFunds();

@Authorized("hasRoles('Manager') && hasAttribute('directReports', employeeId)")
public double getSalary(long employeeId);
```

Jakarta Configuration

- Externalizing application configuration
- Built-in stores
  - Property files, Java system properties, environment variables
- Extensible stores
  - Kubernetes Secrets
  - Secure cloud storage (such as Azure Key Vault)
  - NoSQL stores (such as Azure Redis)
  - Relational database
- `@Inject` into code
- Reference in EL
- Reference in XML
- Moving MicroProfile Config to Jakarta EE



Jakarta Configuration Examples

```
@Inject
@ConfigProperty(name = "admin.group", defaultValue="admin")
private String adminGroup;
```

```
persistence.provider=org.hibernate.ejb.HibernatePersistence
persistence.datasource=java:app/jdbc/CargoTrackerDatabase
persistence.schema.generation.database.action=create
```

```
<data-source>
    <name>java:app/jdbc/CargoTrackerDatabase</name>
    <class-name>org.apache.derby.jdbc.EmbeddedDriver</class-name>
    <url>jdbc:derby:${temp.dir}/cargo-tracker-database</url>
</data-source>
```

Jakarta NoSQL

- Specification for accessing NoSQL databases
- Mapping/template API akin to Jakarta Persistence
- API variants by NoSQL taxonomy
  - Key-value pair, column family, document, and graph
- Repositories, validation, externalized configuration

Jakarta NoSQL Example

```
@Entity public class Order {
    @Id private long id;
    @Column @NotBlank private String description;
    @Column @NotEmpty private List<OrderLine> orderLines;
    @Column @NotNull private Customer customer;
    @Column @NotNull private Address address;
```

```
template.insert(order);
...
List<Order> results = template.select(Order.class)
    .where("description").like("^Pinball").result();

logger.info("Found orders for pinball: " + results.size());
```

Jakarta MVC

- Standard action-based web framework
  - Jakarta Faces continues to evolve separately
  - Standalone, “prospective specification” in Jakarta EE 11
- Model
  - CDI, Validation
- View
  - Facelets, Jakarta Server Pages
- Controller
  - Based on Jakarta REST

Jakarta MVC Example

```
@Controller
@Path("/")
@View("my-index.xhtml")
public class Bookstore {
  @Inject private Models model;
  @Inject private BookService service;
  ...
  @GET
  public void index() {
    ...
    model.put(“books”, books);
  }
}
```

Many Others

- Make persistence.xml optional/empty marker
- JCache as a second-level cache provider

Records and JSON Binding

```
@GET @Path("/shortest-path") @Produces({"application/json"})
public List<TransitPath> findShortestPath(String originUnLocode,
    String destinationUnLocode, String deadline) {
```

```
public record TransitPath (List<TransitEdge> transitEdges) {}

public record TransitEdge (String voyageNumber, String fromUnLocode,
    String toUnLocode, LocalDateTime fromDate, LocalDateTime toDate) {}
```

Job Definition API Example

```
Job job = new JobBuilder(jobName).property("jobk1", "J")
    .listener("jobListener1", new String[]{"jobListenerk1",
        "#{jobParameters['jobListenerPropVal']}"},
    .step(new StepBuilder(stepName)
        .properties(new String[]{"stepk1", "S"}, new String[]{"stepk2", "S"})
    .batchlet(batchlet1Name, new String[]{"batchletk1", "B"},
        new String[]{"batchletk2", "B"})
    .listener("stepListener1", stepListenerProps)
    .stopOn("STOP").restartFrom(stepName).exitStatus()
    .endOn("END").exitStatus("new status for end")
    .failOn("FAIL").exitStatus()
    .nextOn("*").to(step2Name)
    .build())
 ...
.build();
```

Built-In @Service Stereotype

```
@ApplicationScoped
@Transactional
@Lock(READ_WRITE)
@MaxConcurrency(10)
@Monitored // Vendors should add their own sensible functionality
@Stereotype
@Target(TYPE)
@Retention(RUNTIME)
public @interface Service {}
```

Bootstrap API for Jakarta EE

```
public class Application {

    public static void main(String[] args) {
        JakartaApplication.run();
    }
}
```

Beyond Specification Work

- More custom websites for key Jakarta EE technologies
  - Like https://beanvalidation.org, http://www.cdi-spec.org
- Jakarta Starter
  - Helping developers getting started quickly
- Jakarta EE Tutorial
  - The official free resource to learn Jakarta EE
- Jakarta EE Examples
  - Quickly getting working code for most use cases
- Eclipse Cargo Tracker
  - End-to-end application demonstrating architectural practices like Domain-Driven Design

Ways of Contributing

- Follow Jakarta EE technologies that interest you and share opinion
  - https://jakarta.ee/connect/mailing-lists/
- Advocate for a specific change or feature
  - https://jakarta.ee/projects/
- Help implement a change in API, specification, TCK or implementation
  - Sign Eclipse Contributor Agreement
  - https://www.eclipse.org/legal/ECA.php
  - Becoming a committer comes much later
- Engage an Ambassador if needed
  - https://jakartaee-ambassadors.io

Summary

- Jakarta EE 11 almost here, has important changes
- One of the key motivations to move Java EE to Jakarta EE is greater community contribution
- Jakarta EE work is ongoing - time to get involved is now

Resources

- JakartaOne Livestream recordings
  - https://jakartaone.org
- Jakarta EE Community mailing list
  - https://accounts.eclipse.org/mailing-list/jakarta.ee-community
- Jakarta EE X/Twitter handle
  - https://twitter.com/JakartaEE
- Jakarta Tech Talks
  - https://www.meetup.com/jakartatechtalks_