Configuration API JSR

Eclipse MicroProfile Community
Presented by David Blevins, Tomitribe

Focus Areas

- Feed applications with java.util.Property data (Configuration)
- Mix of Internal and External properties (ConfigSources)
- Dependency Injection or Lookup (@ConfigProperty & Config)
- Runtime-change of configuration
- Conversion to any Java data type (Converter)

Simple Example - Referencing Properties

```java
@Dependent
public static class NextFace2Face {

    @Inject
    @ConfigProperty(name="jcp.f2f.location")
    private String place;

    @Inject
    @ConfigProperty(name="jcp.f2f.host")
    private org.jcp.Member host;

    @Inject
    @ConfigProperty(name="jcp.f2f.canceled", defaultValue = "false")
    private Boolean canceled;

    @Inject
    @ConfigProperty(name="jcp.seats.invited")
    private java.util.Optional<Integer> attendees;
```

Simple Example - Programmatic Lookup

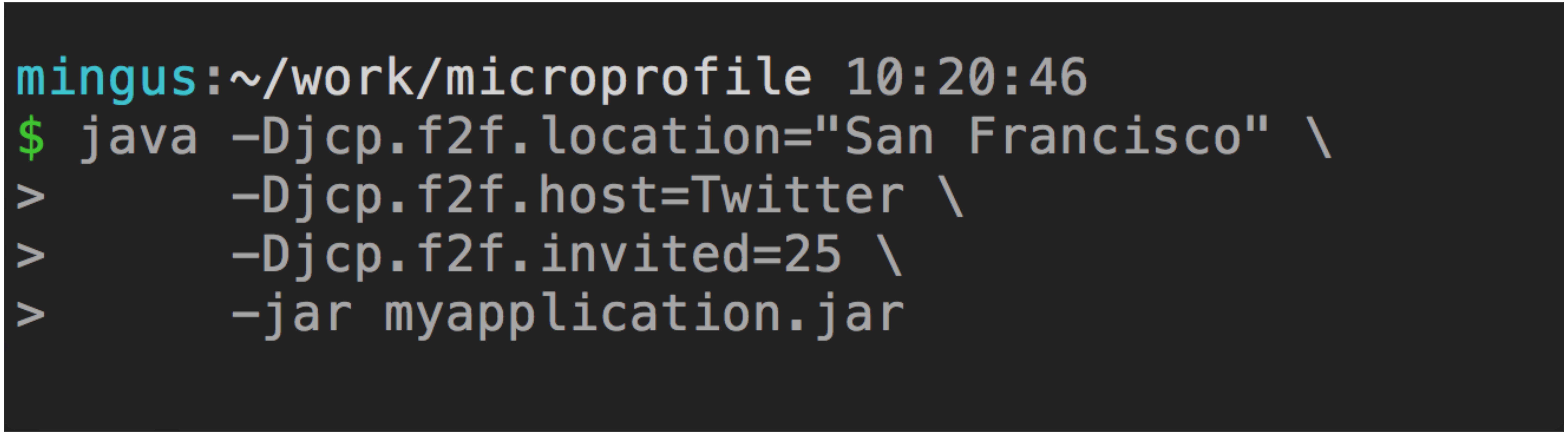
```java
public void feedTheEC(Consumer<Pizza> ec) {

    final Config config = ConfigProviderResolver.instance().getBuilder()
                .addDefaultSources()
                .addDiscoveredSources()
                .withSources(new MyCustomConfigSource())
                .addDiscoveredConverters()
                .build();

    final String location = config.getValue("jcp.f2f.location", String.class);
    final Optional<String> host = config.getOptionalValue("jcp.f2f.host", String.class);
    final Optional<Pizza> pizza = config.getOptionalValue("jcp.f2f.food", Pizza.class);

    pizza.ifPresent(ec);
}
```

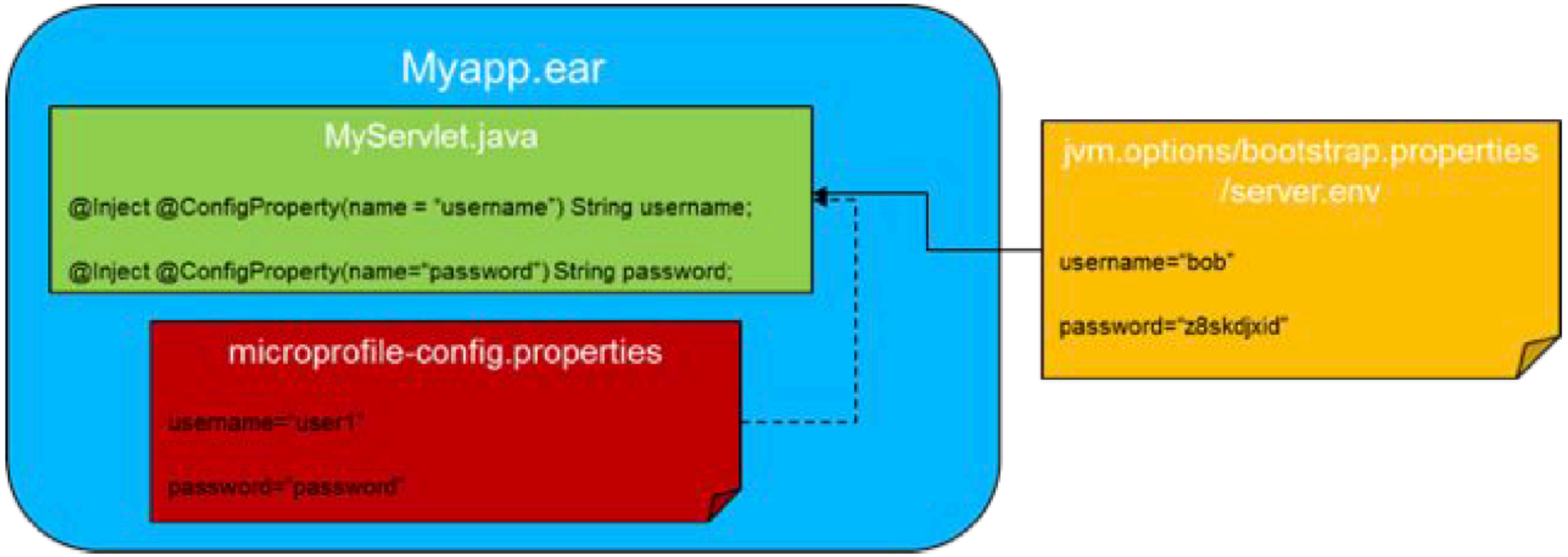
Simple Example - Supplying Properties



```
mingus:~/work/microprofile 10:20:46
$ java -Djcp.f2f.location="San Francisco" \
>      -Djcp.f2f.host=Twitter \
>      -Djcp.f2f.invited=25 \
>      -jar myapplication.jar
```

Built-In ConfigSources

- META-INF/microprofile-config.properties (priority 100)
- System Properties (priority 400)
- Environment variables (priority 300)
- custom properties files (user-chosen priority)



Myapp.ear
MyServlet.java
@Inject @ConfigProperty(name = "username") String username;
@Inject @ConfigProperty(name="password") String password;
microprofile-config.properties
username="user1"
password="password"
jvm.options/bootstrap.properties /server.env
username="bob"
password="z8skdjxid"

Third-Party ConfigSources

- Properties files in the application
- Java System properties
- Java environment variables
- YAML, XML or similar files coerced to properties format
- Database or NoSQL storage
- REST calls to external systems
- Kubernetes environment variables
- Docker properties
- Containerized environments that support environment variables

Influence and History

- DeltaSpike Config (http://deltaspike.apache.org/documentation/configuration.html)
- Extracted parts of DeltaSpike Config (https://github.com/struberg/javaConfig/)
- Apache Tamaya (http://tamaya.incubator.apache.org/)
- Tomitribe Sabot (https://tomitribe.io/p/sabot)
- Apache Geronimo Config (https://svn.apache.org/repos/asf/geronimo/components/config/trunk)
- WebSphere Liberty 2017 Betas (https://developer.ibm.com/wasdev/)

eclipse