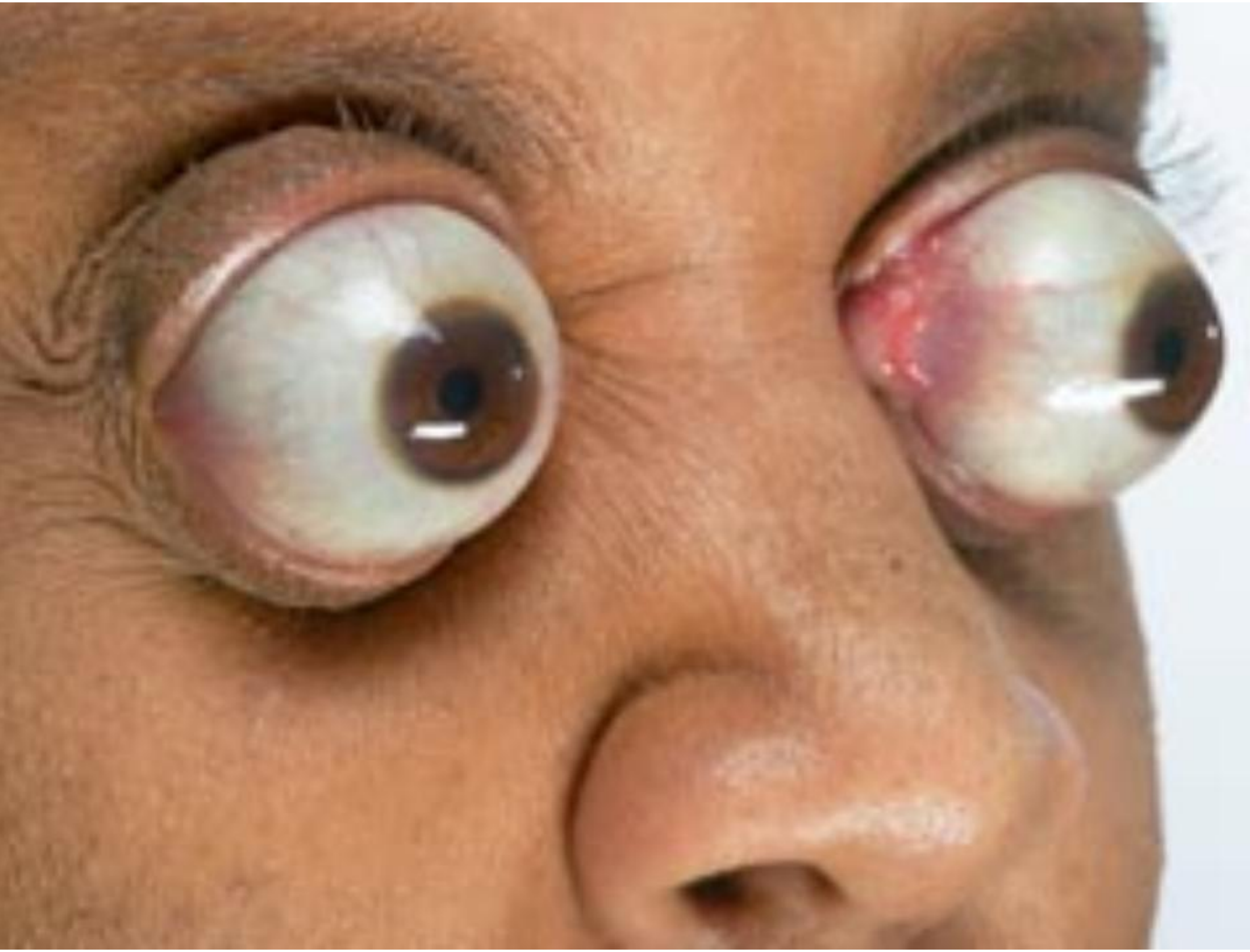
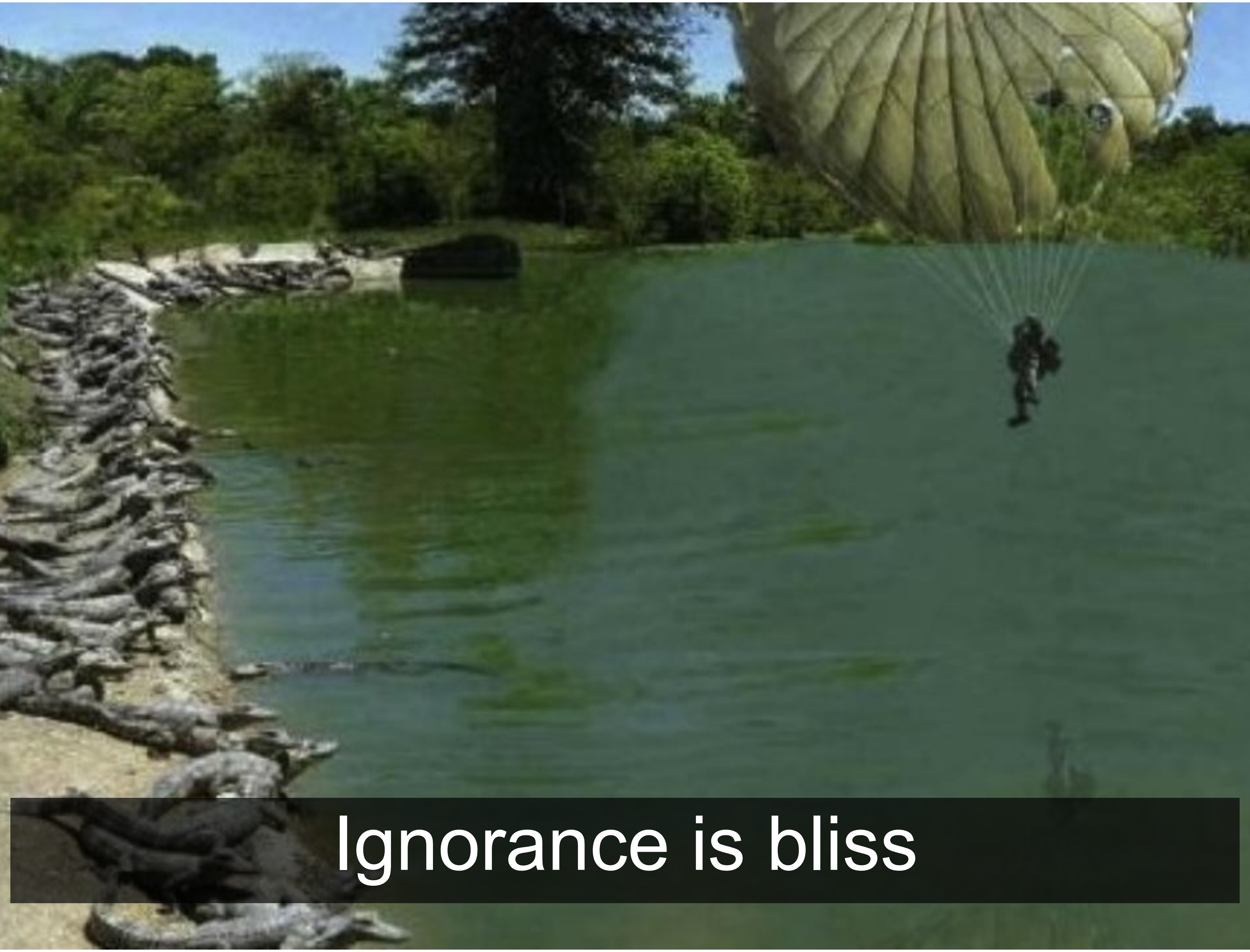
OSGi Best and Worst Practices

Chris Aniszczyk — Red Hat
Jeff McAffer — EclipseSource
Martin Lippert — it-agile
Paul VanderLei — Band XI

Don’t Program OSGi





Ignorance is bliss

It's bad mojo to pollute the POJO

Your code sucks.
Versioning can help.

Good fences make good neighbors

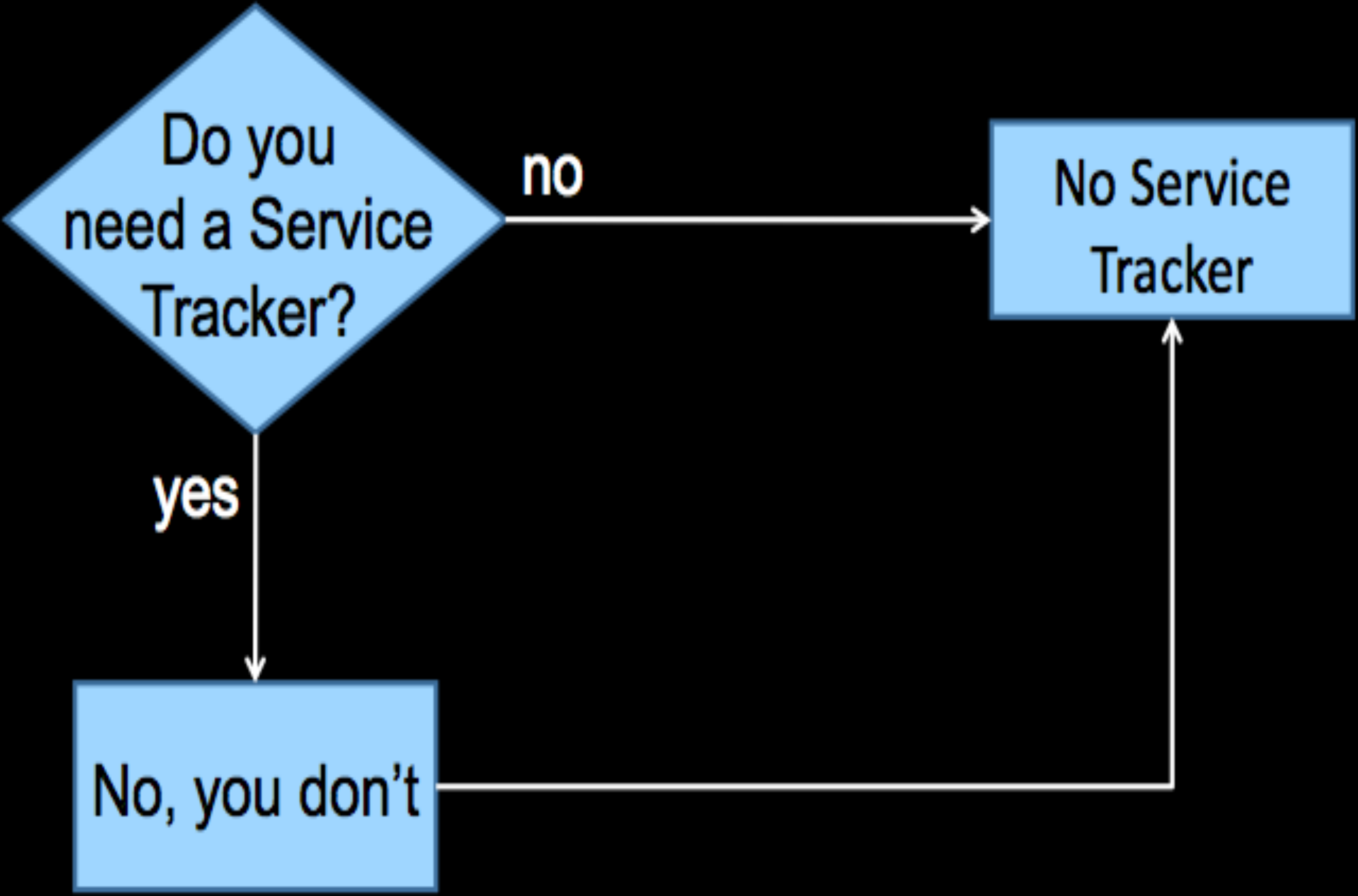
Size Does Matter

Manage your dependencies

Use services

Activator

```
public class Activator implements BundleActivator {
 private BundleContext context;
 private EmergencyMonitor monitor;
 private ServiceTracker gpsTracker;
 private IGps gps;
 private ServiceTracker airbagTracker;
 private IAirbag airbag;

 public void start(BundleContext context) throws Exception {
  this.context = context;
  monitor = new EmergencyMonitor();

  // Start tracking IGps services.
  ServiceTrackerCustomizer gpsCustomizer =
     createGpsCustomizer();
  gpsTracker = new ServiceTracker(context,
     IGps.class.getName(),
   gpsCustomizer);
  gpsTracker.open();

  // Start tracking IAirbag services.
  ServiceTrackerCustomizer airbagCustomizer =
   createAirbagCustomizer();
  airbagTracker = new ServiceTracker(context,
     IAirbag.class.getName(),
   airbagCustomizer);
  airbagTracker.open();
 }

 public void stop(BundleContext context) throws Exception {
  // Stop tracking IAirbag services.
  airbagTracker.close();

  // Stop tracking IGps services.
  gpsTracker.close();
 }

 private ServiceTrackerCustomizer createGpsCustomizer() {
  return new ServiceTrackerCustomizer() {
   public Object addingService(ServiceReference reference) {
    Object service  = context.getService(reference);
    synchronized (this) {
     if (Activator.this.gps == null) {
      Activator.this.gps = (IGps) service;
      Activator.this.bind();
     }
    }
    return service;
   }

   public void removedService(
     ServiceReference reference, Object service) {
    synchronized (this) {
     if (service != Activator.this.gps)
      return;
     Activator.this.unbind();
     Activator.this.bind();
    }
   }

   public void modifiedService(ServiceReference reference,
     Object service) {
    // No service property modifications to handle.
   }
  };
 }

 private ServiceTrackerCustomizer createAirbagCustomizer() {
  return new ServiceTrackerCustomizer() {
   public Object addingService(ServiceReference reference) {
    Object service  = context.getService(reference);
    synchronized (this) {
     if (Activator.this.airbag == null) {
      Activator.this.airbag = (IAirbag) service;
      Activator.this.bind();
     }
    }
    return service;
   }

   public void removedService(
     ServiceReference reference, Object service) {
    synchronized (this) {
     if (service != Activator.this.airbag)
      return;
     Activator.this.unbind();
     Activator.this.bind();
    }
   }

   public void modifiedService(ServiceReference referenc
     Object service) {
    // No service property modifications to handle.
   }
  };
 }

 private void bind() {
  if (gps == null) {
   gps = (IGps) gpsTracker.getService();
   if (gps == null)
    return; // No IGps service.
  }
  if (airbag == null) {
   airbag = (IAirbag) airbagTracker.getService();
   if (airbag == null)
    return; // No IAirbag service.
  }
  // Bind IGps and IAirbag to the EmergencyMonitor
  monitor.bind(gps, airbag);
 }

 private void unbind() {
  if (gps == null || airbag == null)
   return;
  monitor.unbind();
  gps = null;
  airbag = null;
 }
}
```


Do you need a Service Tracker?
no
No Service Tracker
yes
No, you don't

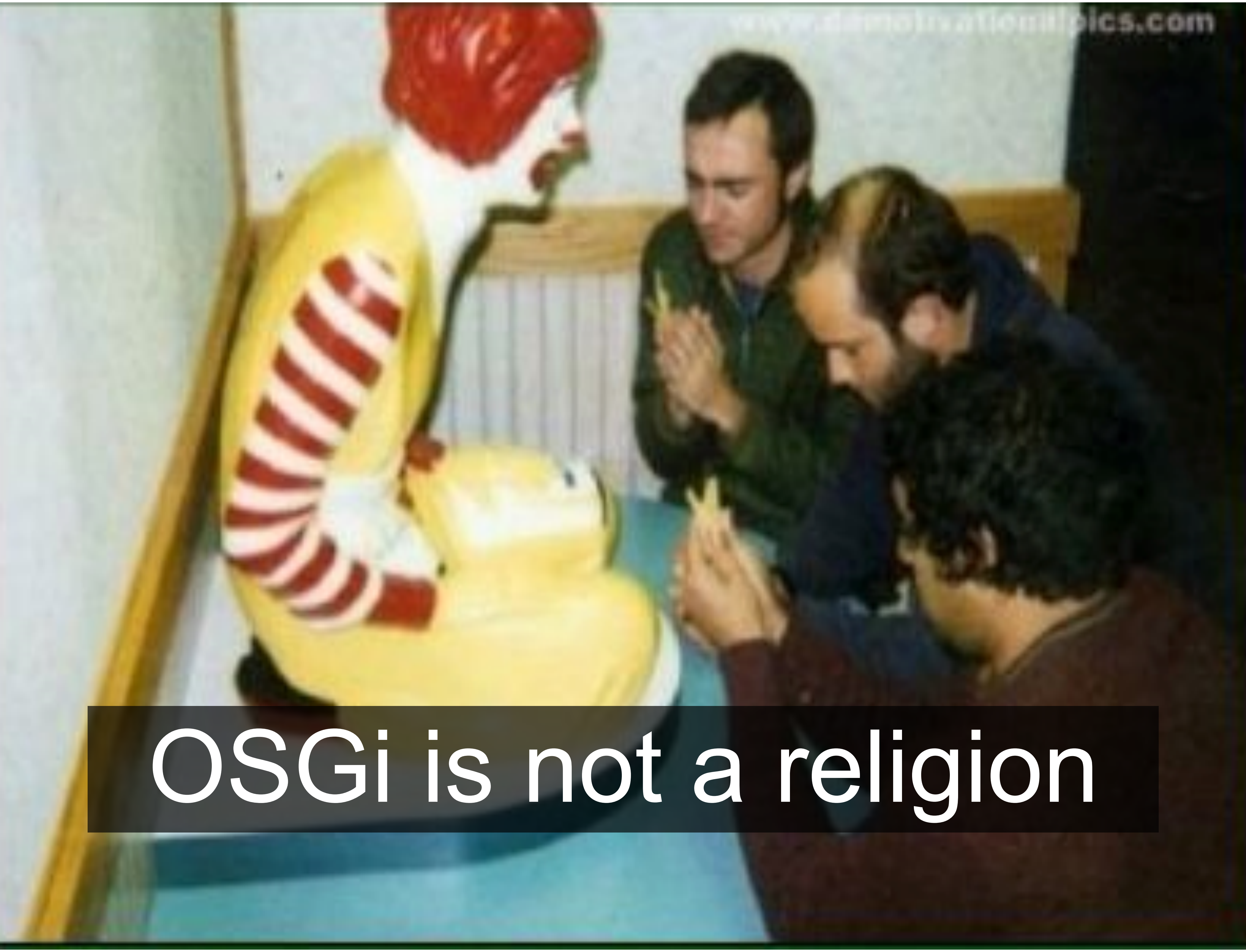
Not everything should be a service

...and OSGi made all the apps dynamic

The Easter Bunny is dead

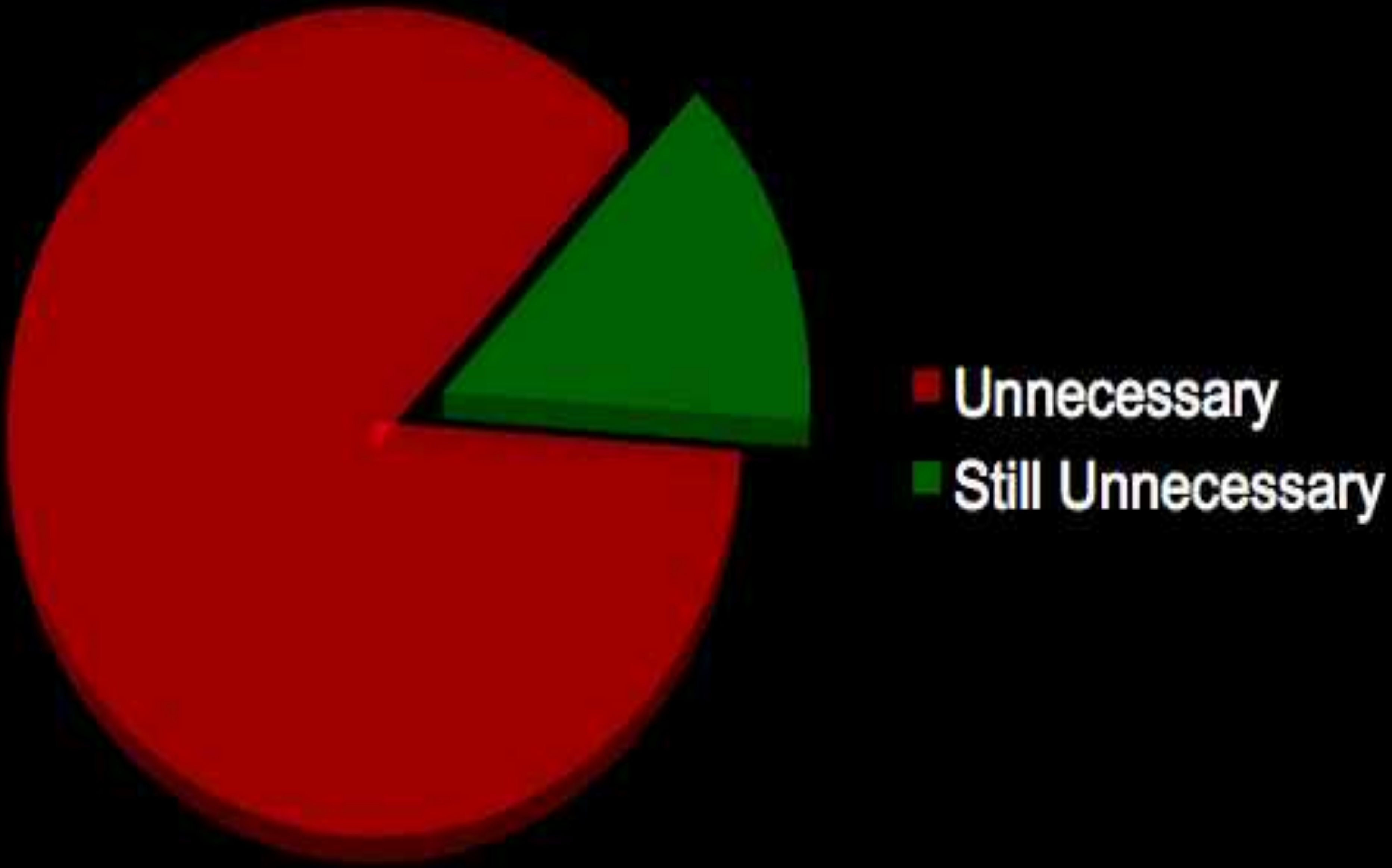
Free!

If you don't test it,
it doesn't work!



OSGi is not a religion

Need for Wire Admin



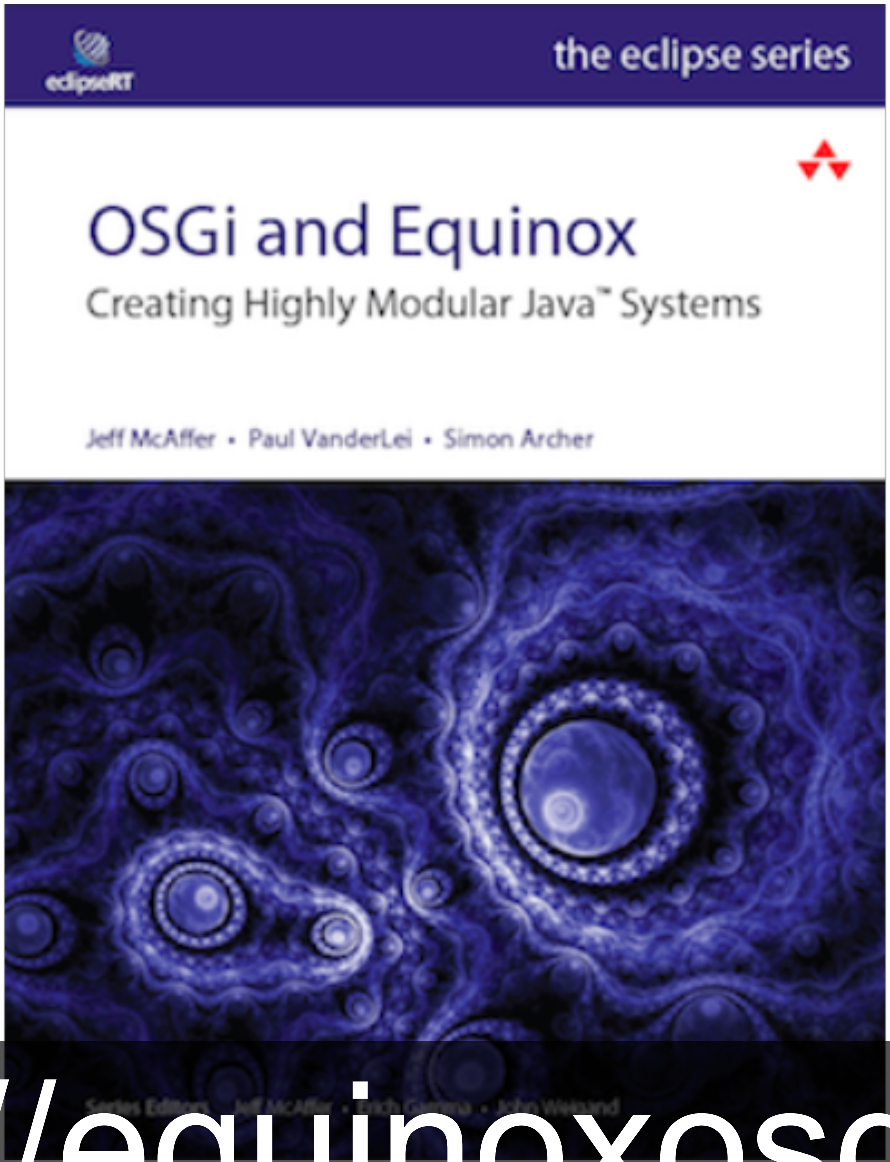
Now some controversy

Require-Bundle

Import-Package

Services or Extensions?

Got Questions?


eclipseRT
the eclipse series
OSGi and Equinox
Creating Highly Modular Java™ Systems
Jeff McAffer • Paul VanderLei • Simon Archer
Series Editors Jeff McAffer • Erich Gamma • John Wiegand

http://equinoxosgi.org