ISCAS 中国科学院软件研究所
Institute of Software, Chinese Academy of Sciences

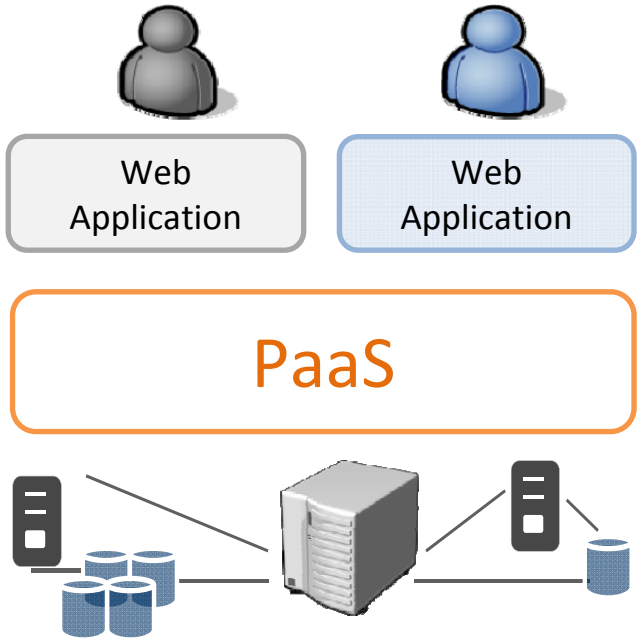
Once Access Controller

Heng Wu

Technology Center of Software Engineering
Institute of Software
Chinese Academy of Sciences

Background

- PaaS is a kind of typical cloud service mode.
- Tenants share resources, such as File, Storage, etc.
- Tenants' data is usually processed remotely in unknown machines that tenants do not operate.



Customizability requirements(1)

- PaaS provider, such as Google AppEngine, can customize the insecurity classes.

JDK Version	Google App Engine White list [1]	Total public classes in JDK	Total classes in JDK
Sun JDK 1.6.0	1313	5206	7069

Table1 Classes are allowed in google AppEngine

Customizability requirements(2)

- PaaS provider can customize the sensitive operations.
- For example, the operation java.net.Inet4Address.getByName(host), which is security insensitive, will not be allowed to invoke in Google AppEngine .

```
public class Class {

    public void operate() {
        // TODO
    }
}
```

Security insensitive operation

```
public class Class {

    public void operate() {
        System.getSecurityManager();
        // TODO
    }
}
```

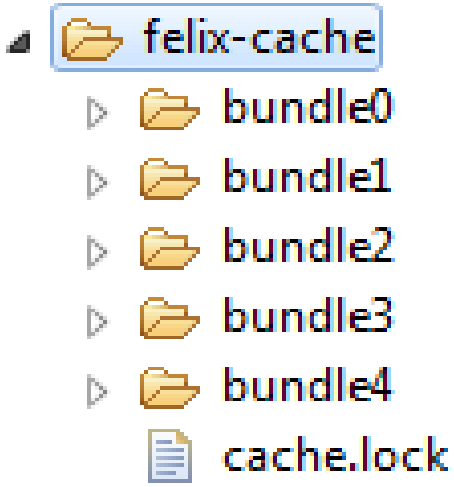
Security sensitive operation

Availability requirements

- Scalability is a key feature of PaaS.
- Web applications are stateless in Google AppEngine.
- Google AppEngine does not support local file operation which may lead to state inconsistent.
- Unfortunately, when one try to cache an Image from Blobstore with ImageService, it is fail.

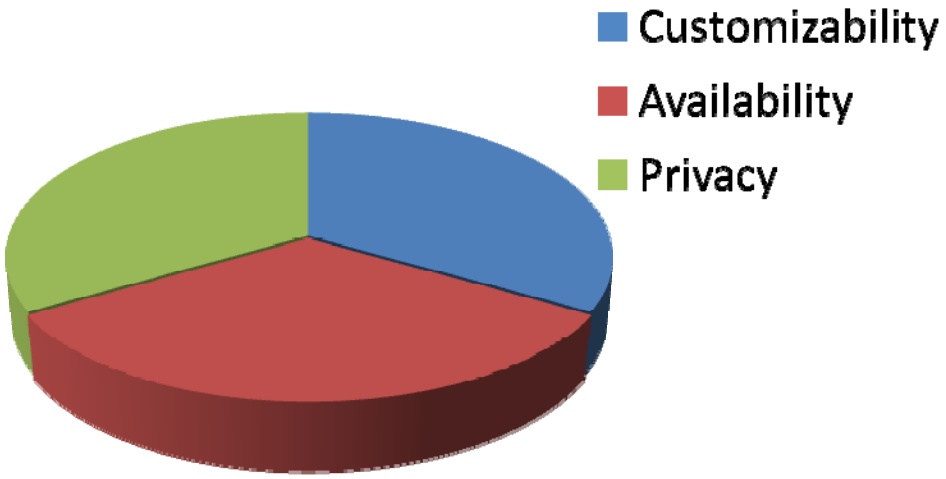
Privacy requirements

- Multi-tenants is a key feature of PaaS , which led to the tenant privacy protection problems.
- For example, in the OSGi Specification, All bundles have File Permission for the bundle persistent storage area, it is insecurity.



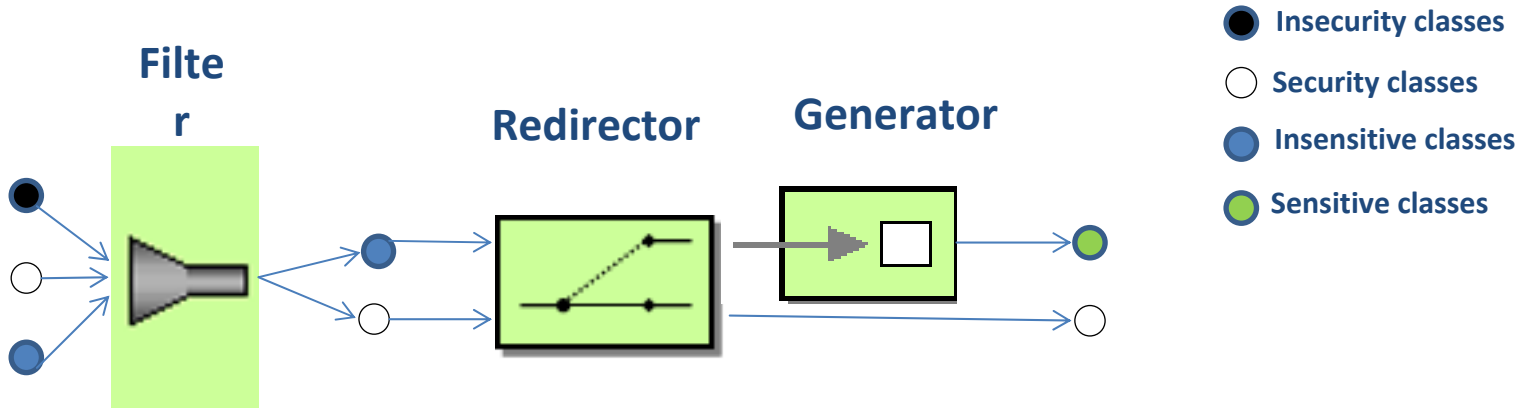
Summary

- Motivation
  - How to meet customizability requirements.
  - How to meet availability requirements.
  - How to meet privacy requirements.



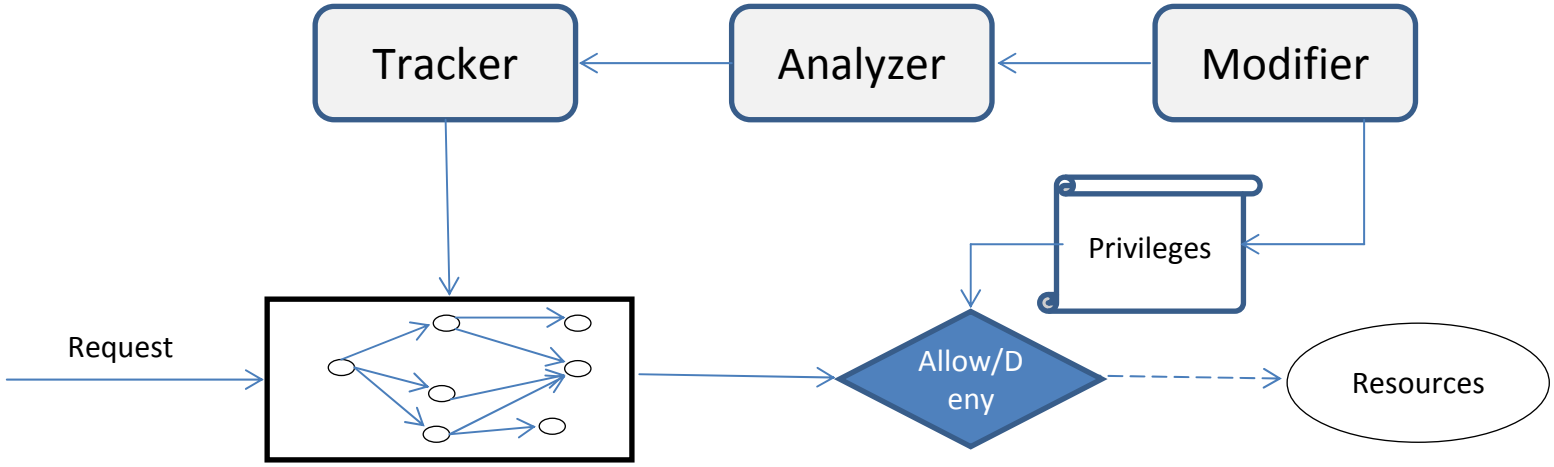
Our current work(1)

- Sensitive classes and operations customizer
  - Filter: override ClassLoader with blacklist mechanism for insecurity classes
  - Generator: generate proxy classes with sensitive operations automatically.
  - Redirector: override ClassLoader with redirection mechanism for sensitive operations.



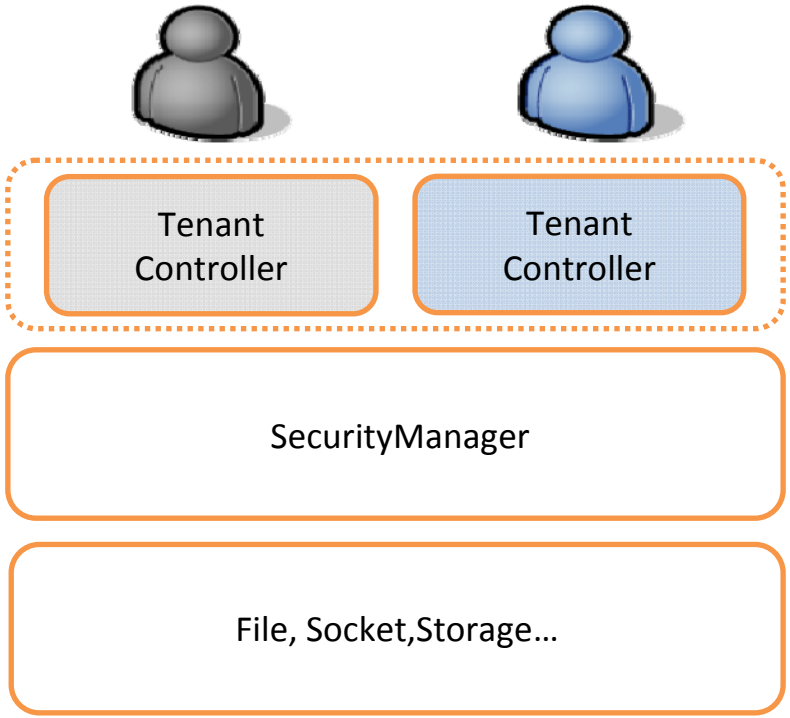
Our current work (2)

- Context-aware Authenticator
  - Tracker : trace the path into the application component architecture.
  - Modifier: modify the web application's privileges at runtime.
  - Analyzer: analyze web application whether we need to modify privileges.



Our current work (3)

- Privilege separator
  - Implemented as an extension of Java2 Security Model
  - Access Control based on instance isolation.



Future Work

- Performance Optimization
  - **Some authentication results are irrelevant with the input.**
  - Consider:
    - Cache the authentication result
- Policies Composition
  - The composition of access control policies is the key to determine access control policies for distributed aggregated resource

What can we do in OW2 Cloudware

- Extending the OSGi security services
  - Support the principle of least privilege
  - Support the sensitive classes and operations customization
  - Support Context-aware authentication
  - Support fine-grained tenant-sensitive separation

Thanks! Q & A

Our current work (1)

- An Access Control Model
  - Implemented as an extension of Java2 Security Model
  - Access Control based on instance isolation and context-aware authentication
  - Features
    - Automatically generate strategy file
    - Context-aware authentication