



Using JASON
to Secure
SOA

Lukasz
Chmielewski,
Richard
Brinkman
Jaap-Henk
Hoepman,
Bert Bos

Concepts

Jason
SOA

JASON and
SOA

Architecture
JASON in SOA
Generation
Source Code
Annotations
WS-* Transition
Class Loaders

Plans

Conclusion

Using JASON to Secure SOA

1st International Workshop on Middleware Security
MidSec 2008, Leuven, Belgium

Łukasz Chmielewski Richard Brinkman
Jaap-Henk Hoepman Bert Bos

Digital Security Group
Radboud University Nijmegen

02-12-2008



Contents

Using JASON
to Secure
SOA

Lukasz
Chmielewski,
Richard
Brinkman
Jaap-Henk
Hoepman,
Bert Bos

Concepts

Jason
SOA

JASON and
SOA

Architecture
JASON in SOA
Generation
Source Code
Annotations
WS-* Transition
Class Loaders

Plans

Conclusion

- 1 Concepts
 - Jason
 - Service Oriented Architecture
- 2 JASON and SOA
 - JASON architecture
 - JASON within SOA
 - Generation process
 - Example Source Code
 - JASON annotations
 - Transition to WS-* standards
 - Multiple Class Loaders
- 3 Plans
- 4 Conclusion



Concepts

Jason

Using JASON
to Secure
SOA

Lukasz
Chmielewski,
Richard
Brinkman
Jaap-Henk
Hoepman,
Bert Bos

Concepts

Jason
SOA

JASON and
SOA

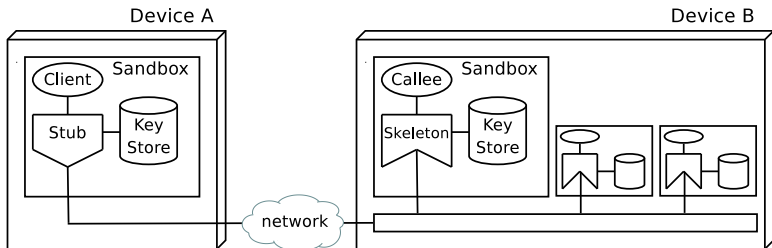
Architecture
JASON in SOA
Generation
Source Code
Annotations
WS-* Transition
Class Loaders

Plans

Conclusion

Jason provides a **middleware** (which was originally designed for smartcard systems) that offers a secure environment in which

- Applications run safely in their **sandboxes**.
- Communicate securely with **Secure Communication Layer (SCL)**.
- Developer specifies security expectations, **JASON compiler** produces the actual secure implementation.





Concepts

Service Oriented Architecture

Using JASON to Secure SOA

Lukasz
Chmielewski,
Richard
Brinkman
Jaap-Henk
Hoepman,
Bert Bos

Service Broker

Client

Service

Concepts

Jason
SOA

JASON and SOA

Architecture
JASON in SOA
Generation
Source Code
Annotations
WS-* Transition
Class Loaders

Plans

Conclusion



Concepts

Service Oriented Architecture

Using JASON
to Secure
SOA

Lukasz
Chmielewski,
Richard
Brinkman
Jaap-Henk
Hoepman,
Bert Bos

Concepts

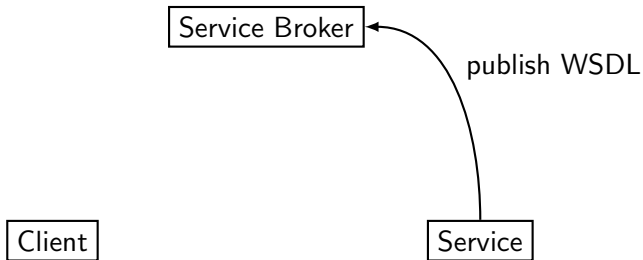
Jason
SOA

JASON and
SOA

Architecture
JASON in SOA
Generation
Source Code
Annotations
WS-* Transition
Class Loaders

Plans

Conclusion





Concepts

Service Oriented Architecture

Using JASON
to Secure
SOA

Lukasz
Chmielewski,
Richard
Brinkman
Jaap-Henk
Hoepman,
Bert Bos

Concepts

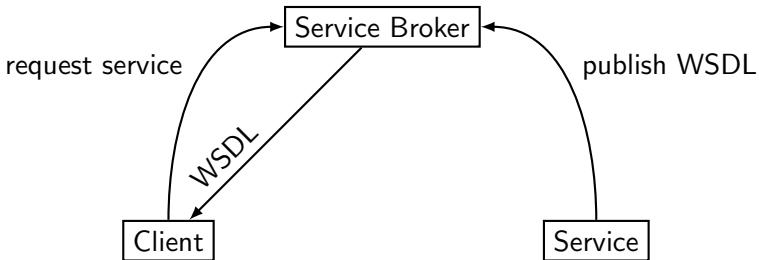
Jason
SOA

JASON and
SOA

Architecture
JASON in SOA
Generation
Source Code
Annotations
WS-* Transition
Class Loaders

Plans

Conclusion





Concepts

Service Oriented Architecture

Using JASON
to Secure
SOA

Lukasz
Chmielewski,
Richard
Brinkman
Jaap-Henk
Hoepman,
Bert Bos

Concepts

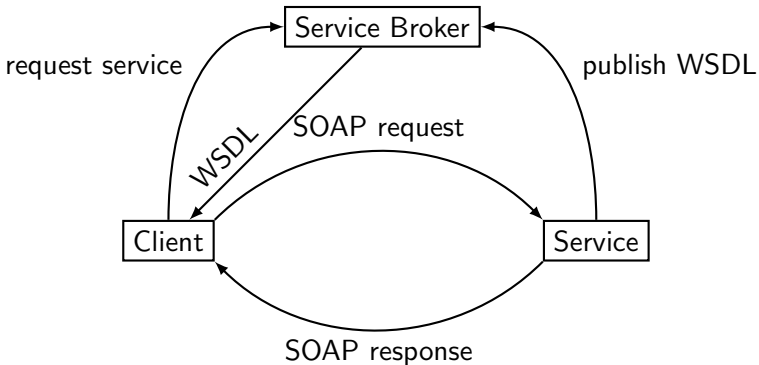
Jason
SOA

JASON and
SOA

Architecture
JASON in SOA
Generation
Source Code
Annotations
WS-* Transition
Class Loaders

Plans

Conclusion





JASON architecture

Using JASON
to Secure
SOA

Lukasz
Chmielewski,
Richard
Brinkman
Jaap-Henk
Hoepman,
Bert Bos

Concepts

Jason
SOA

JASON and
SOA

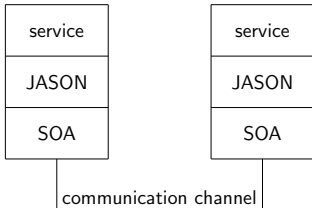
Architecture

JASON in SOA
Generation
Source Code
Annotations
WS-* Transition
Class Loaders

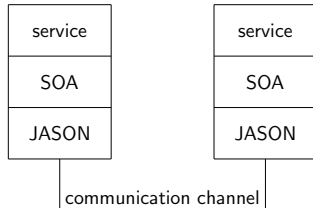
Plans

Conclusion

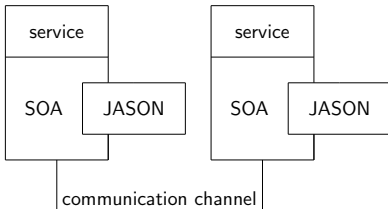
JASON on top of SOA:



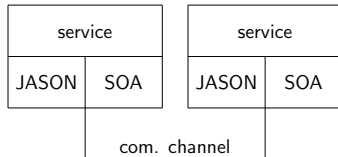
SOA on top of JASON:



SOA plugged into JASON:



SOA next to JASON:





JASON within SOA

JASON framework

Using JASON to Secure SOA

Lukasz Chmielewski,
Richard Brinkman
Jaap-Henk Hoepman,
Bert Bos

Concepts

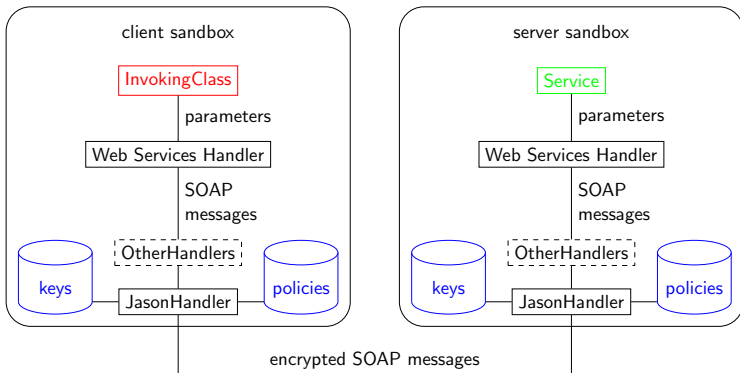
Jason
SOA

JASON and SOA

Architecture
JASON in SOA
Generation
Source Code
Annotations
WS-* Transition
Class Loaders

Plans

Conclusion





Generation process

Jason within SOA

Using JASON
to Secure
SOA

Lukasz
Chmielewski,
Richard
Brinkman
Jaap-Henk
Hoepman,
Bert Bos

Concepts

Jason
SOA

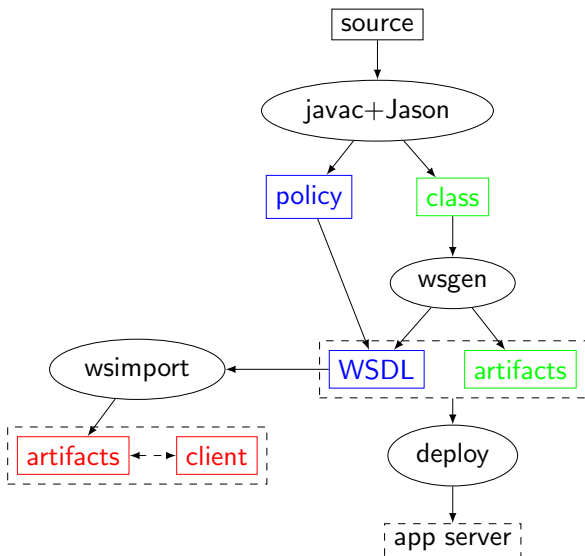
JASON and
SOA

Architecture
JASON in SOA

Generation
Source Code
Annotations
WS-* Transition
Class Loaders

Plans

Conclusion





Example Source Code

Example 1

Using JASON
to Secure
SOA

Lukasz
Chmielewski,
Richard
Brinkman
Jaap-Henk
Hoepman,
Bert Bos

Concepts

Jason
SOA

JASON and
SOA

Architecture
JASON in SOA
Generation

Source Code

Annotations
WS-* Transition
Class Loaders

Plans

Conclusion

jason/test/HomeEmergencyDoor.java

```
package jason.test;
import jason.annotation.*;
import javax.jws.*;
@WebService
@Roles({"OWNER", "AMBULANCE"})
public class HomeEmergencyDoor {
    @WebMethod
    @AccessibleTo({"OWNER"})
    @Authentic
    public boolean changeDoorStatus
        (@Confidential @Authentic boolean status) {
        ...
    }
}
```



Example Source Code

Example 2

Using JASON
to Secure
SOA

Lukasz
Chmielewski,
Richard
Brinkman
Jaap-Henk
Hoepman,
Bert Bos

Concepts

Jason
SOA

JASON and
SOA

Architecture
JASON in SOA

Generation

Source Code

Annotations

WS-* Transition

Class Loaders

Plans

Conclusion

```
jason/test/HomeEmergencyDoor.java
```

```
...
@AvailablePolicies(" OldVersion", " NewVersion")
public class HomeEmergencyDoor {
    @WebMethod
    @Policies({
        @Policy(name=" OldVersion",
            accessibleTo=@AccessibleTo({" AMBULANCE" })),
        @Policy(name=" NewVersion",
            accessibleTo=@AccessibleTo({" AMBULANCE" })),
        logged=@Logged,
        authentic=@Authentic
    })
    public boolean openDoor () {
        ...
    }
}
```



JASON annotations

Using JASON to Secure SOA

Lukasz
Chmielewski,
Richard
Brinkman
Jaap-Henk
Hoepman,
Bert Bos

Concepts

Jason
SOA

JASON and SOA

Architecture
JASON in SOA
Generation
Source Code
Annotations
WS-* Transition
Class Loaders

Plans

Conclusion

The class annotations:

- 1 **@Roles** (String[] list)
- 2 **@AccessibleTo** (String[] list)
- 3 **@AvailablePolicies** (String[] list)

The method annotations:

- 1 **@AccessibleTo** (String[] list)
- 2 **@Logged**

The parameters / result annotations:

- 1 **@Confidential** (String[] encryptedBy); field: encryptedBy
- 2 **@Authentic** (String[] signedBy); field: signedBy
- 3 **@Integrity**
- 4 **@Fresh** ??

@Policies({ **@Policy** (String name, **@Annotation** annotation, ...), ... })



Transition to WS-* standards

Using JASON
to Secure
SOA

Lukasz
Chmielewski,
Richard
Brinkman
Jaap-Henk
Hoepman,
Bert Bos

Concepts

Jason
SOA

JASON and
SOA

Architecture
JASON in SOA

Generation
Source Code
Annotations

WS-* Transition
Class Loaders

Plans

Conclusion

annotation	security standard
@Roles	SAML, WS-Trust, WS-SecureConversation
@AccessibleTo	SAML, WS-Trust, WS-SecureConversation
@AvailablePolicies	WS-Policy, WS-SecurityPolicy
@Policy	WS-Policy, WS-SecurityPolicy
@Logged	
@Confidential	WS-Security, XML Encrypt
@Authentic	WS-Security, XML Signature
@Integrity	WS-Reliability
@Fresh	WS-Security



Multiple Class Loaders

JavaCard Class Loader

Using JASON
to Secure
SOA

Lukasz
Chmielewski,
Richard
Brinkman
Jaap-Henk
Hoepman,
Bert Bos

Concepts

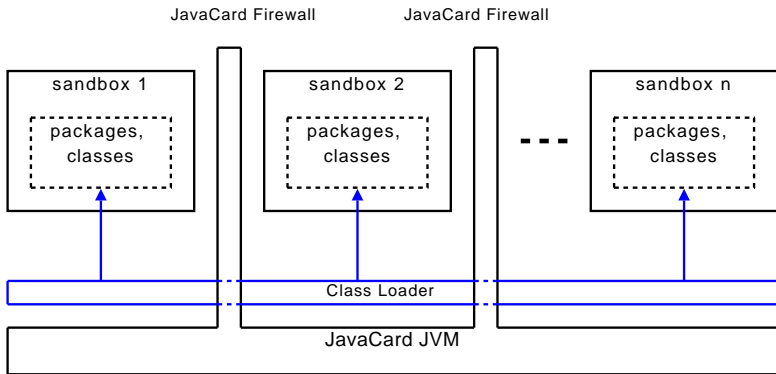
Jason
SOA

JASON and
SOA

Architecture
JASON in SOA
Generation
Source Code
Annotations
WS-* Transition
Class Loaders

Plans

Conclusion





Multiple Class Loaders

Java SE Class Loaders

Using JASON
to Secure
SOA

Lukasz
Chmielewski,
Richard
Brinkman
Jaap-Henk
Hoepman,
Bert Bos

Concepts

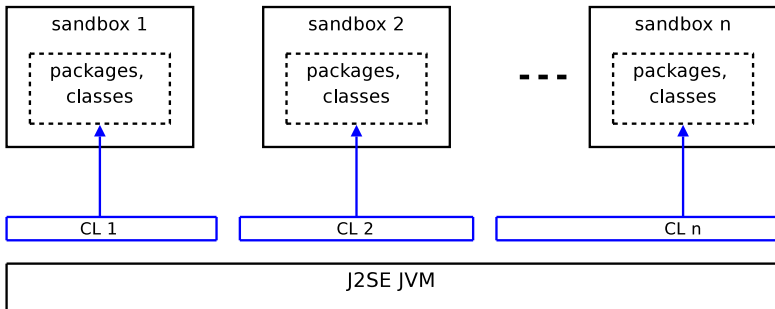
Jason
SOA

JASON and
SOA

Architecture
JASON in SOA
Generation
Source Code
Annotations
WS-* Transition
Class Loaders

Plans

Conclusion





Plans

Using JASON to Secure SOA

Lukasz
Chmielewski,
Richard
Brinkman
Jaap-Henk
Hoepman,
Bert Bos

Concepts

Jason
SOA

JASON and SOA

Architecture
JASON in SOA
Generation
Source Code
Annotations
WS-* Transition
Class Loaders

Plans

Conclusion

- going on with implementation: JASON compiler and JASON framework
- key management: XML Key Registration Service, and XML Key Information Service; implemented in e.g., OpenXKMS
- delegation



Conclusion

Using JASON to Secure SOA

Lukasz
Chmielewski,
Richard
Brinkman
Jaap-Henk
Hoepman,
Bert Bos

Concepts

Jason
SOA

JASON and SOA

Architecture
JASON in SOA
Generation
Source Code
Annotations
WS-* Transition
Class Loaders

Plans

Conclusion

- JASON adds security to SOA / Web Services
- JASON eases programming SOA security
- JASON within SOA – combination of approaches
- JASON fits within Java / JAAS / SOA / Web Services mainstream technology
- Representing JASON interface using Java annotations



Questions

Using JASON
to Secure
SOA

Lukasz
Chmielewski,
Richard
Brinkman
Jaap-Henk
Hoepman,
Bert Bos

Concepts

Jason
SOA

JASON and
SOA

Architecture
JASON in SOA
Generation
Source Code
Annotations
WS-* Transition
Class Loaders

Plans

Conclusion

?



Delegation

Using JASON
to Secure
SOA

Lukasz
Chmielewski,
Richard
Brinkman
Jaap-Henk
Hoepman,
Bert Bos

Concepts

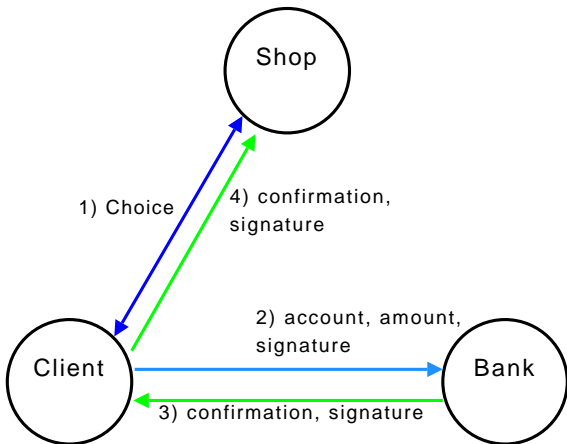
Jason
SOA

JASON and
SOA

Architecture
JASON in SOA
Generation
Source Code
Annotations
WS-* Transition
Class Loaders

Plans

Conclusion



- forwarding / delegation



Delegation

Authentic case

Using JASON to Secure SOA

Lukasz
Chmielewski,
Richard
Brinkman
Jaap-Henk
Hoepman,
Bert Bos

Concepts

Jason
SOA

JASON and SOA

Architecture
JASON in SOA
Generation
Source Code
Annotations
WS-* Transition
Class Loaders

Plans

Conclusion

```
public class C1 {
    public static void main(String[] args) {
        stubC2.m2(2);
    }
}

public class C2 {
    public void m2(@Authentic(signedBy="C1") int x) {
        stubC3.m3(x);
    }
}

public class C3 {
    public void m3(@Authentic(signedBy="C1") int y) {}
}
```



Delegation

Confidential case

Using JASON to Secure SOA

Lukasz
Chmielewski,
Richard
Brinkman
Jaap-Henk
Hoepman,
Bert Bos

Concepts

Jason
SOA

JASON and SOA

Architecture
JASON in SOA
Generation
Source Code
Annotations
WS-* Transition
Class Loaders

Plans

Conclusion

```
public class C1 {
    public static void main(String[] args) {
        stubC2.m2(2);
    }
}

public class C2 {
    public void m2(@Confidential(encryptedBy="C1") byte[] x) {
        stubC3.m3(x);
    }
}

public class C3 {
    public void m3(@Confidential(encryptedBy="C1") int y) {};
}
```



Delegation

Confidential case

Using JASON to Secure SOA

Lukasz
Chmielewski,
Richard
Brinkman
Jaap-Henk
Hoepman,
Bert Bos

Concepts

Jason
SOA

JASON and SOA

Architecture
JASON in SOA
Generation
Source Code
Annotations
WS-* Transition
Class Loaders

Plans

Conclusion

```
public class C1 {  
    public static void main(String[] args) {  
        stubC2.m2(2);  
    }  
}  
  
public class C2 {  
    public void m2(@Token(realtype="int") byte[] x) {  
        stubC3.m3(x);  
    }  
}  
  
public class C3 {  
    public void m3(@Confidential(encryptedBy="C1") int y) {}  
}
```