

Secure Java applications

Johan Peeters

<http://www.johanpeeters.com>

independent software architect

Objectives

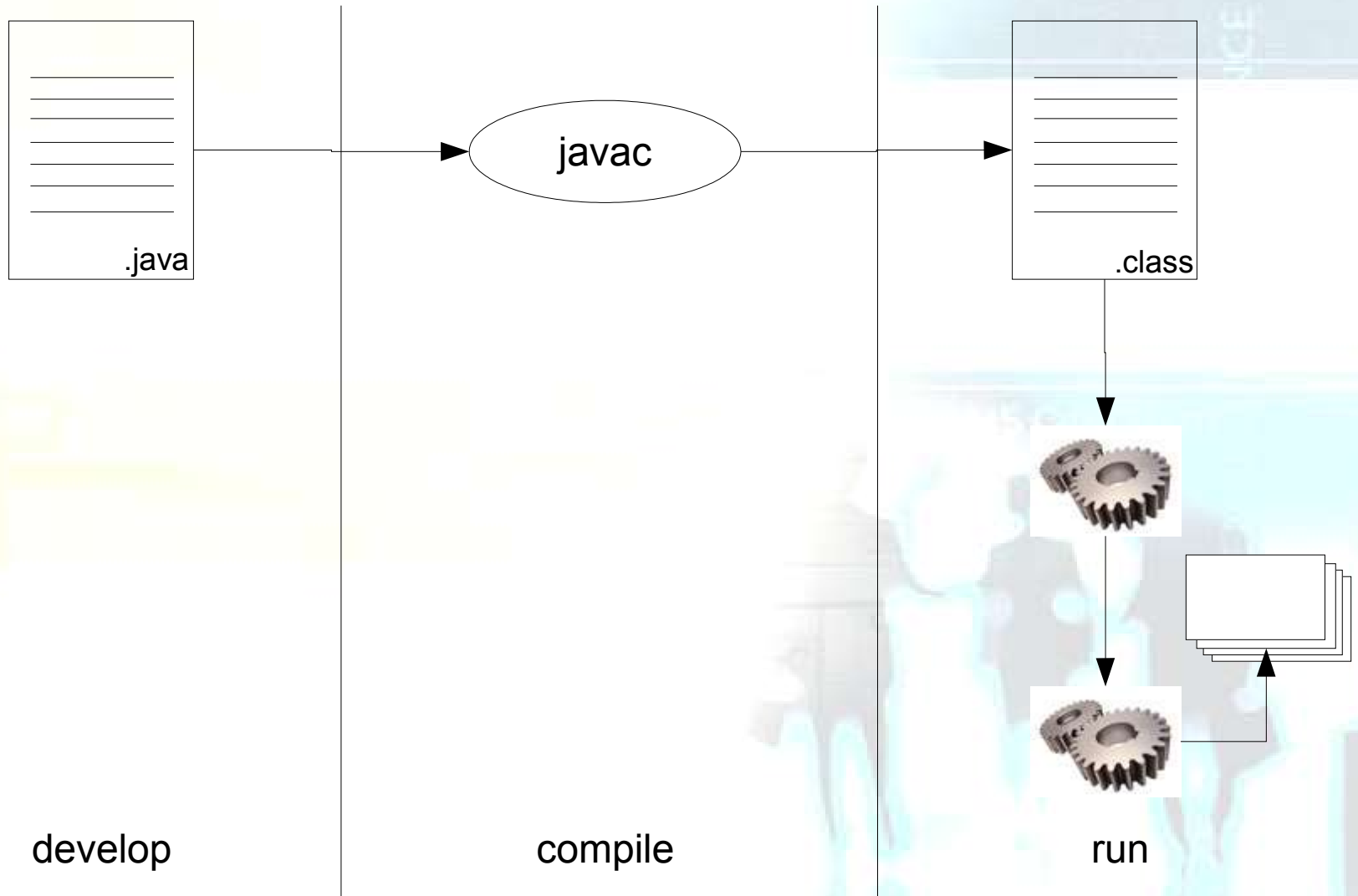
- Explain
 - Access control
 - Class file verification
- Assess goodness of fit
 - Threats resulting from running untrusted code
 - Confrontation with resource limitations
 - Are threats tackled appropriately?

Out of scope

Java eases the pain in secure application development

- Bugs are less likely than in C/C++ because of
 - Memory management
 - Garbage collection
 - No pointer arithmetic
 - Type soundness
- Security APIs

Overview

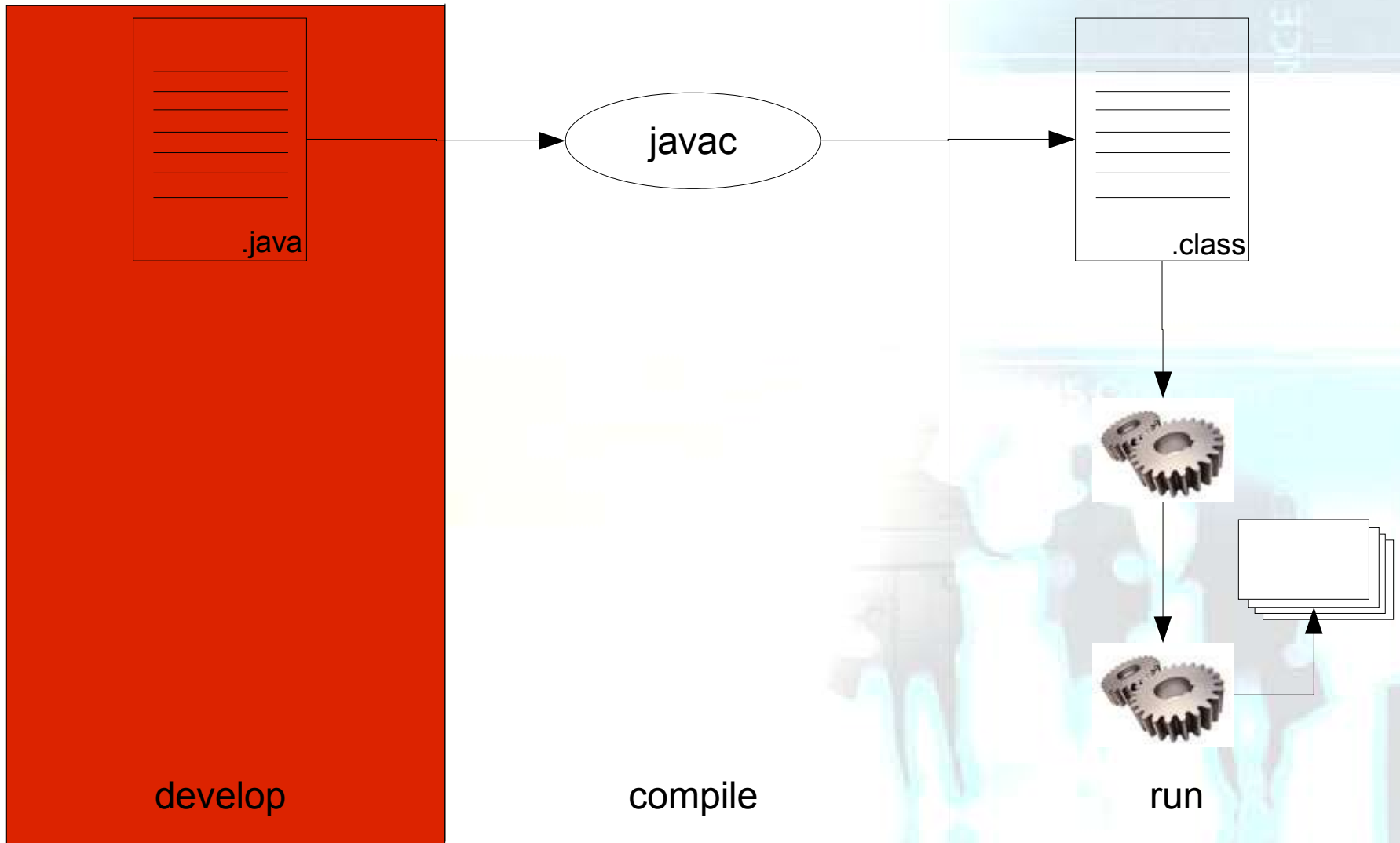


develop

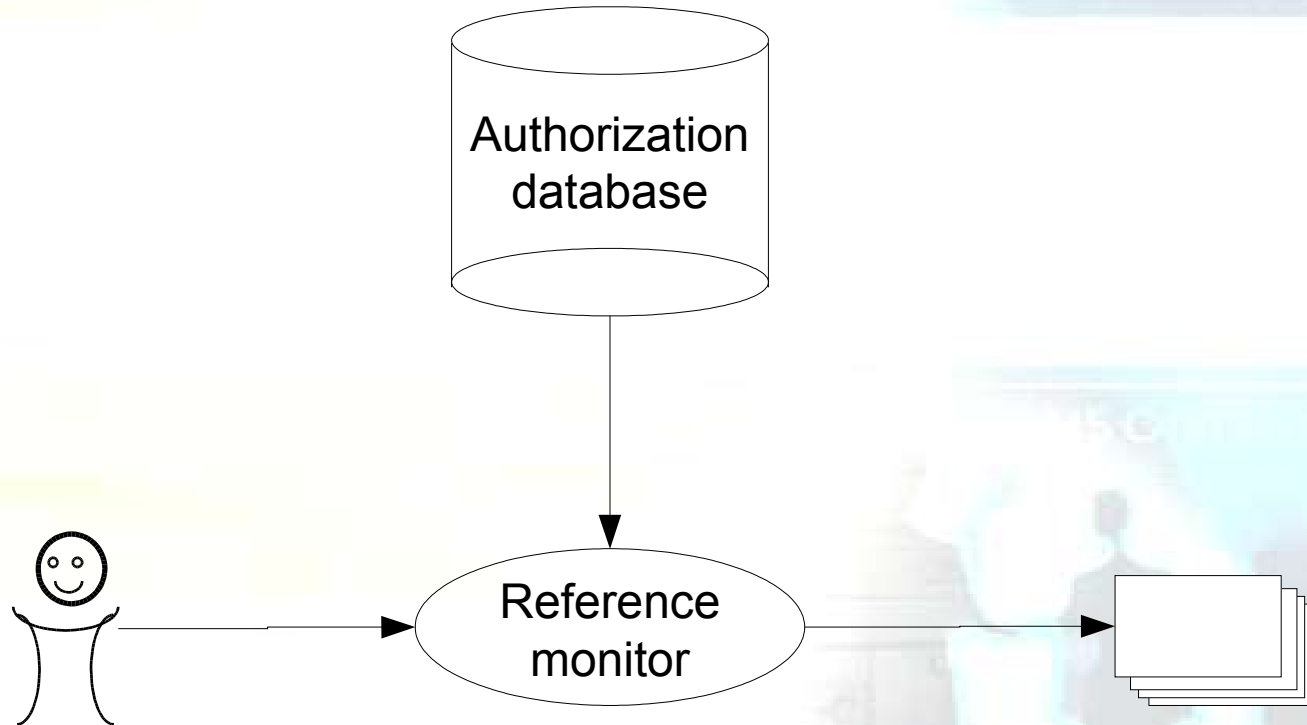
compile

run

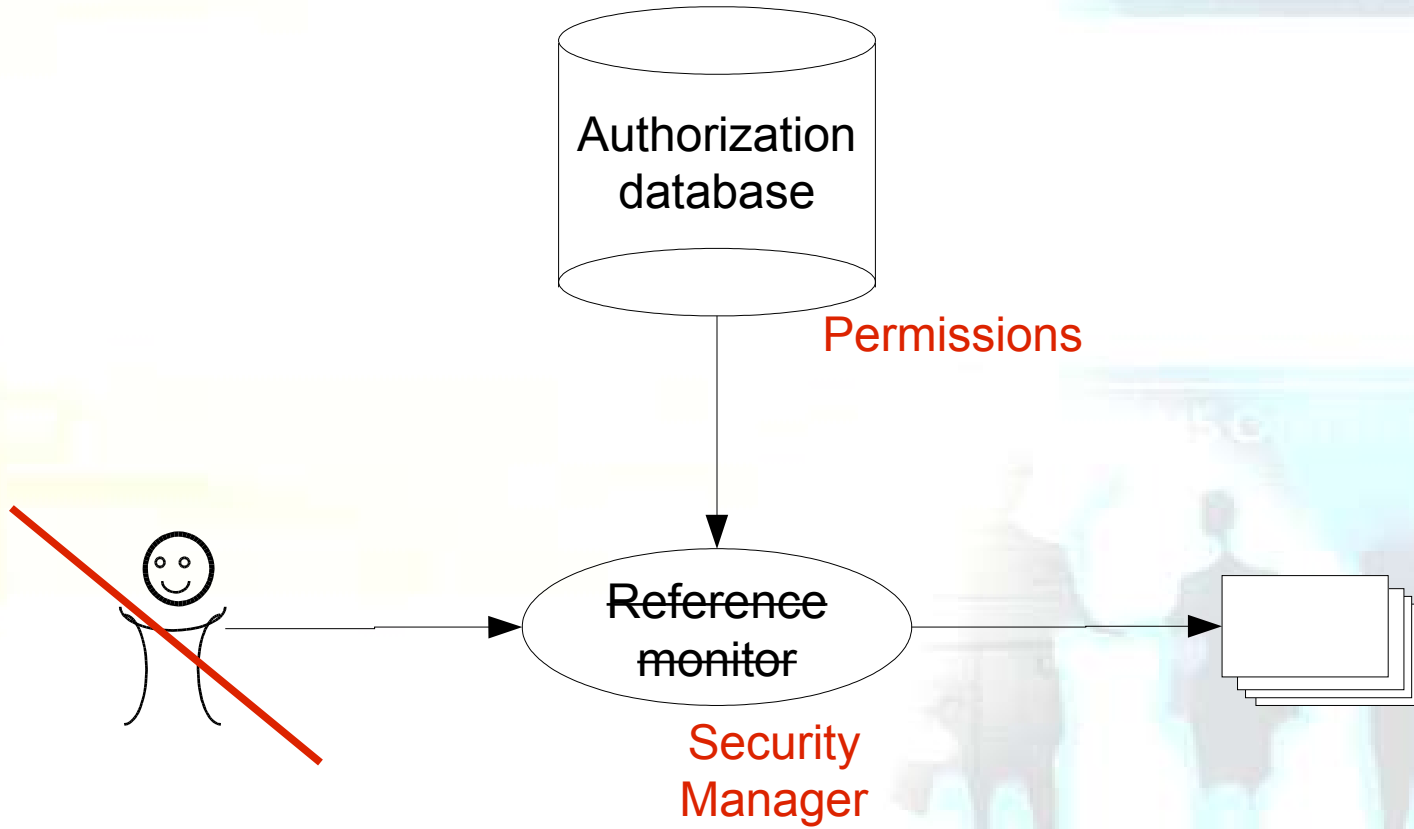
Trust breaks down I



Reference Monitor



Access control in Java

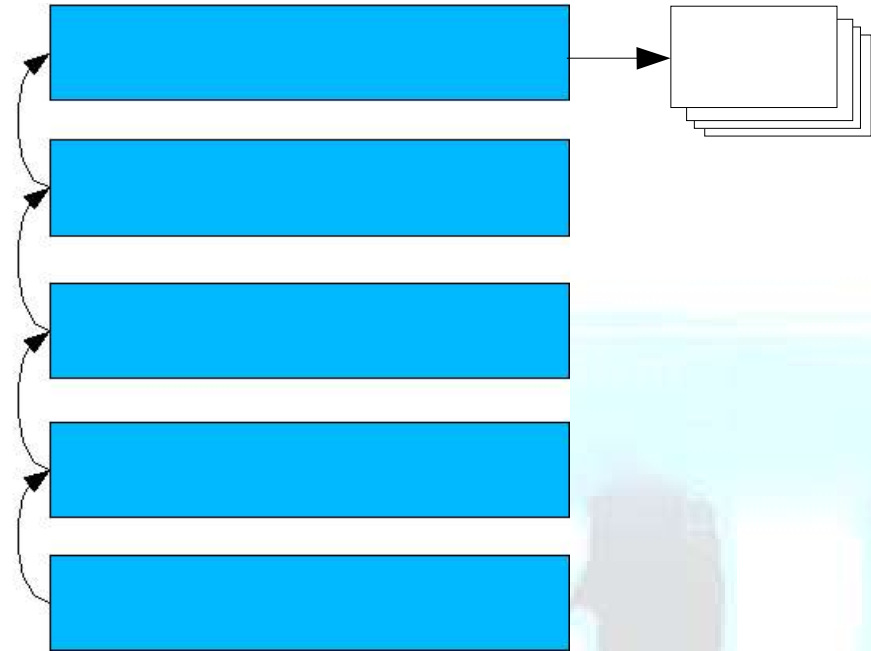


Policy vs mechanism

```
grant codebase "https://www.johanpeeters.com/apps" {
    permission java.io.FilePermission "/home/yo" "read,
        write";
}
grant signedBy "Johan Peeters" {
    permission java.io.FilePermission "/home/yo" "read,
        write";
}
grant principal com.johanpeeters.UserPrincipal "yo" {
    permission java.io.FilePermission "/home/yo" "read,
        write";
}
```

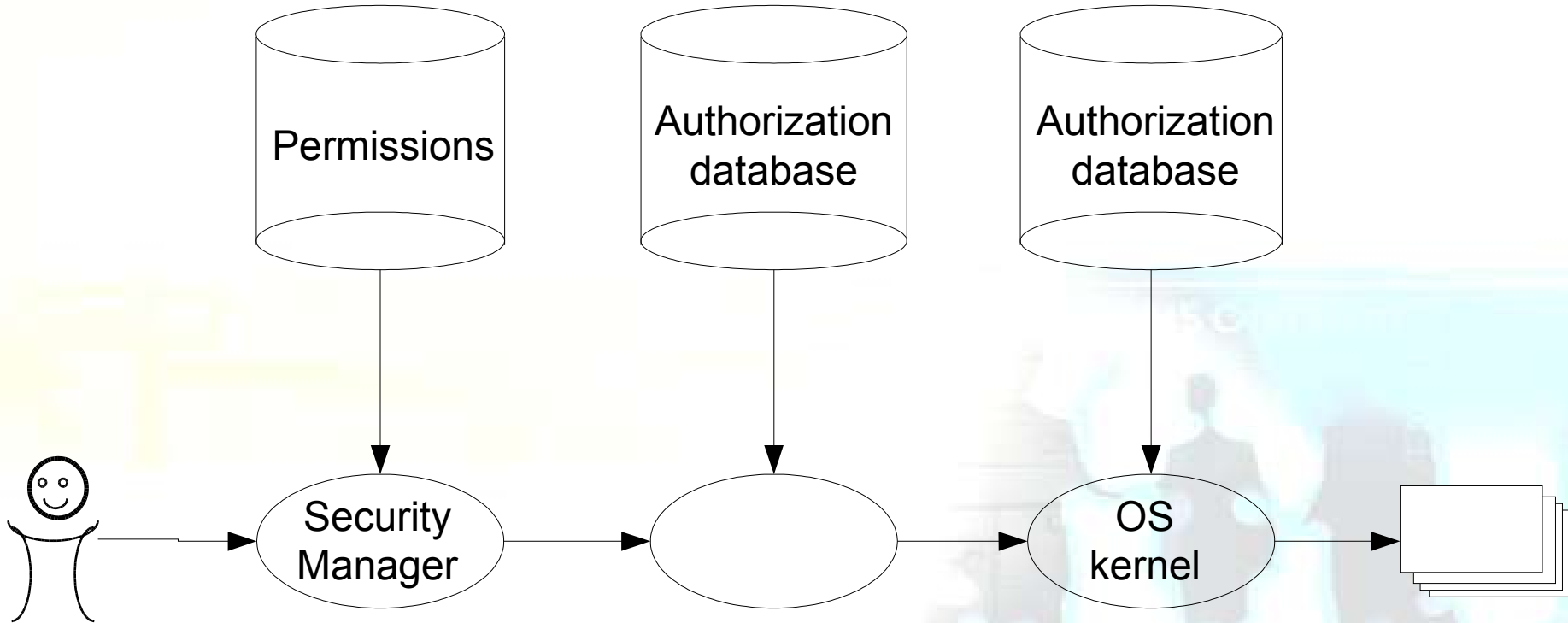
Stack inspection

```
package com.johanpeeters;
...
public class FileWriter
    extends ...{
    public void write(...)
        ...
    }
package com.attacker;
import com.johanpeeters.*;
...
FileWriter fw = new
    FileWriter();
fw.write(...)
...
```



- Permission granted or denied on the basis of the call stack
- Expensive

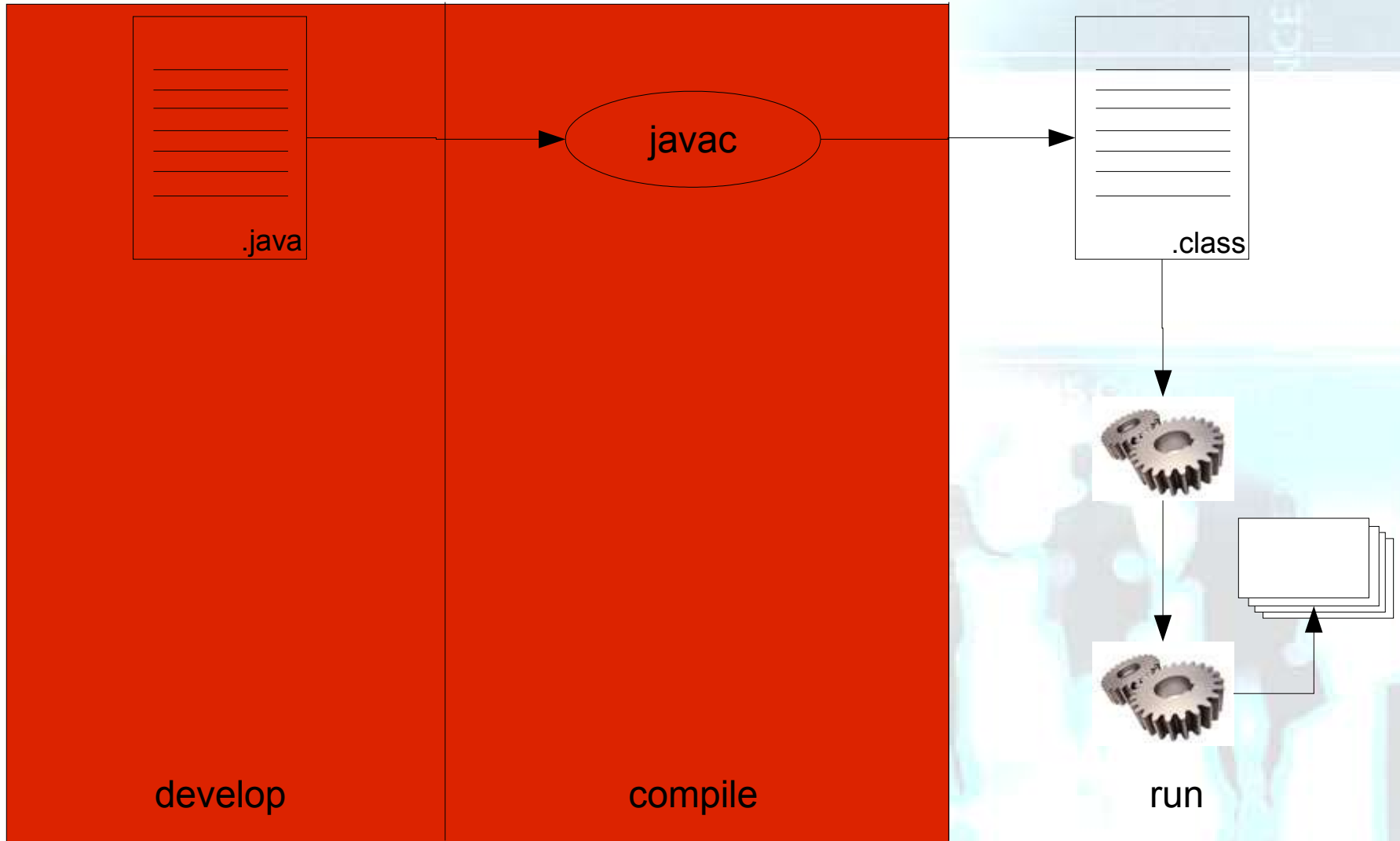
Access control duplication



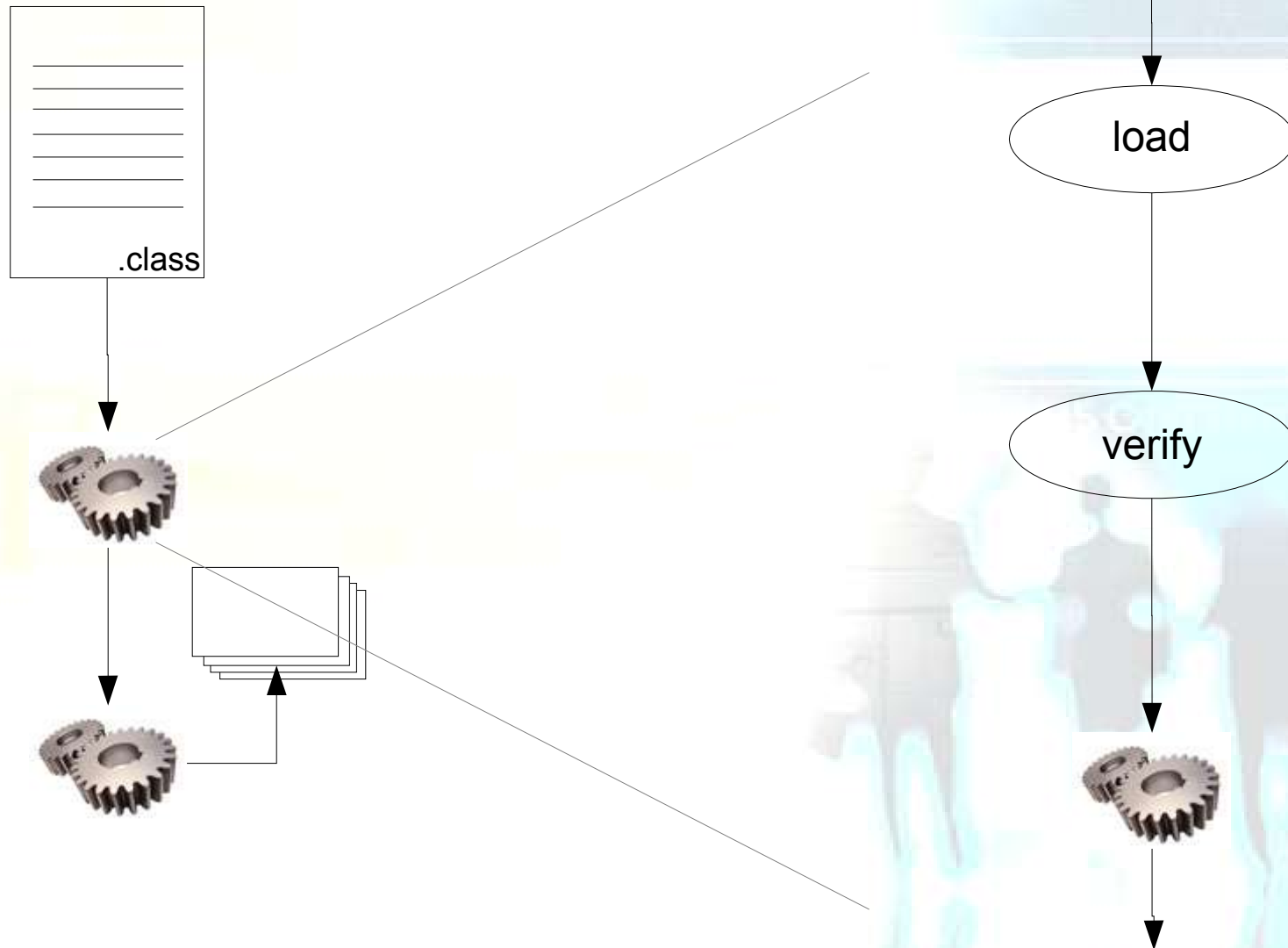
Simplicity

- If implications of security policies become intractable, people will be lax about security configuration
- Do the simplest possible thing, but no simpler:
 - All access control by OS?
 - Today's operating systems have little or no notion of code origin
 - Java's notion of 'user' different from the OS'
- Is the future brighter?
 - Java as an OS
 - IDS, IPS, ... become aware of code base
 - Hardware/OS support for checking code certification

Trust breaks down II



.class file verification



Verification on embedded Java

- Verification takes much time and space
- preverification
 - does not extend trusted computing base
 - technique akin to proof-carrying code

Concluding ...

	Mandatory	Recurring overhead
Access control	No	Yes
Class file verification	Yes	No

- DoS attacks
- Difference between a flaw and a bug
 - Both result in vulnerabilities, i.e. potential exploits
 - Java eliminates certain classes of bugs
 - It does nothing for the flaws

Resources

- Li Gong, Gary Ellison and Mary Dageforde, *Inside Java 2 Platform Security*, second edition, Addison Wesley, 2003
- <http://wiki.javapolis.com>
- <http://www.secure-application-development.org> or <http://www.secappdev.org>

Questions and answers

Any questions?