

Cool Concurrency with CheckThread

Joe Conti

CheckThread.org



List Example

```
private final List mylist = new ArrayList();  
  
// Called by multiple threads  
public void mymethod(Object obj)  
{  
    mylist.add(obj);  
}
```

List Example

```
private final List mylist = new ArrayList();
```

```
// Called by multiple threads
```

```
public void mymethod(Object obj)
{
    mylist.add(obj);
}
```



ArrayList.add(...) is not thread safe!



List Example with a Threading Policy

```
private final List mylist = new ArrayList();  
  
// Called by multiple threads  
@ThreadSafe  
public void mymethod(Object obj)  
{  
    mylist.add(obj);  
}
```

Thread Policy



DEMO:

Catching List Concurrency Bug At Compile Time

CheckThread's View of the World

```
@ThreadSafe
```

```
methodWantsToBeThreadSafe(...)
```

```
{
```

```
    sharedData.notThreadSafe(...)
```

```
}
```



Danger!

SWT Example

```
private Button myButton;  
  
// Runs on main thread  
public void mymethod()  
{  
    myButton.setBounds(0,0,100,100);  
}
```

SWT Example

```
private Button myButton;  
  
// Runs on main thread  
public void mymethod()  
{  
    myButton.setBounds(0,0,100,100);  
}
```









SWT widgets should only be invoked on the UI thread.

DEMO:

Catching an SWT
Thread Confinement Violation
at Compile Time

Three Thread Policy Contracts

		Client	Implementer
@ThreadSafe	Can invoke shared data method concurrently on multiple threads		
@NotThreadSafe	Client can invoke shared data method from one thread at a time.		
@ThreadConfined	Invoke method on a specific runtime thread.		

Using XML Instead of Annotations

Equivalent

Foo.java:

```
class Foo {  
    @ThreadSafe  
    public void someMethod(Object obj)  
    { ... }  
}
```

threadpolicy.xml:

```
<class pattern="com.Foo">  
    <policy type="ThreadSafe">  
        <method pattern="some.*" />  
    </policy>  
</class>
```

Primitive Data Example

```
private int val;  
  
// Called by multiple threads  
public void mymethod()  
{  
    val = 1;  
    int foo = 1/val;  
    val = 0;  
}
```

Primitive Data Example

```
private int val;
```

```
// Called by multiple threads
```

```
public void mymethod()
```

```
{
```

```
    val = 1;
```

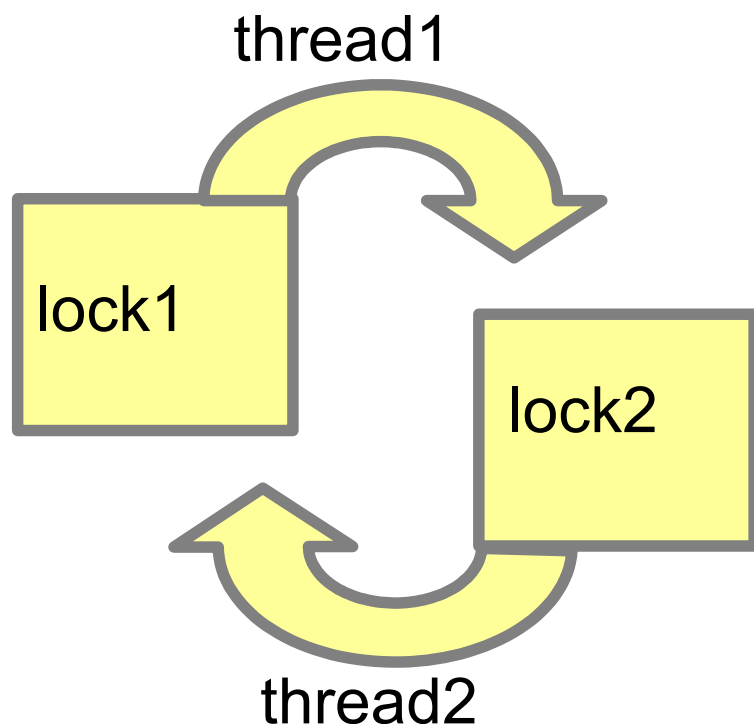
```
    int foo = 1/val;
```

```
    val = 0;
```

```
}
```

**write + read + no synchronization
= possible race condition**

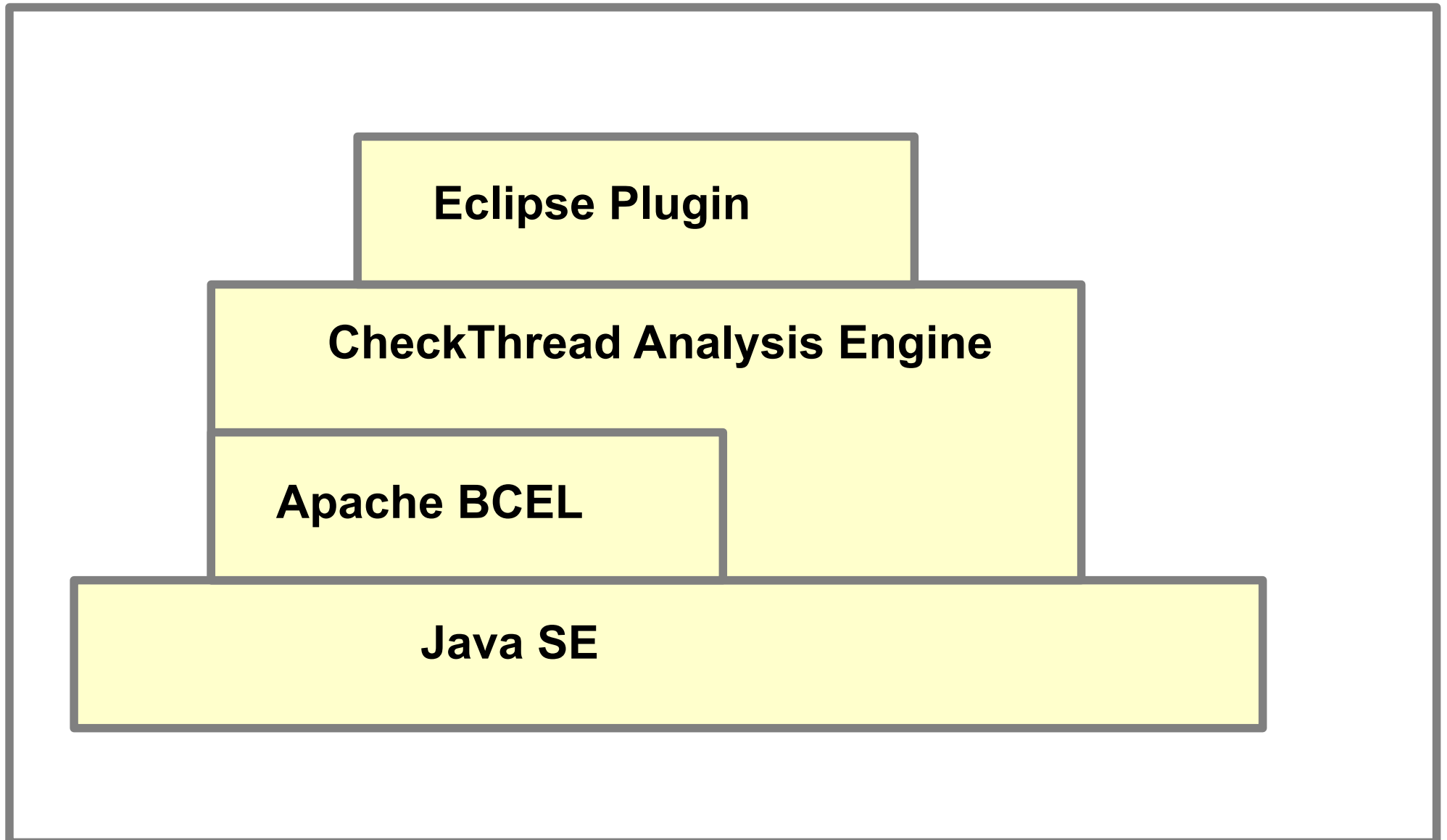
Lock Loop Detection



```
thread1:  
synchronized(lock1) {  
    synchronized(lock2) {}  
}
```

```
thread2:  
synchronized(lock2) {  
    synchronized(lock1) {}  
}
```

CheckThread Architecture



CheckThread Next Steps...

- Gather feedback from Eclipse-Java community
- Relax annotation/XML markup requirement
- Add more concurrency analysis and bug detection

Further Reading about Java Concurrency

- “Java Concurrency in Practice”
 - Brian Goetz
 - This book introduces `@ThreadSafe` and `@NotThreadSafe`
- “Java Threads”
 - Scott Oaks & Henry Wong

For info, downloads, & source code:

Site: <http://checkthread.org>

Contact: checkthread@gmail.com

