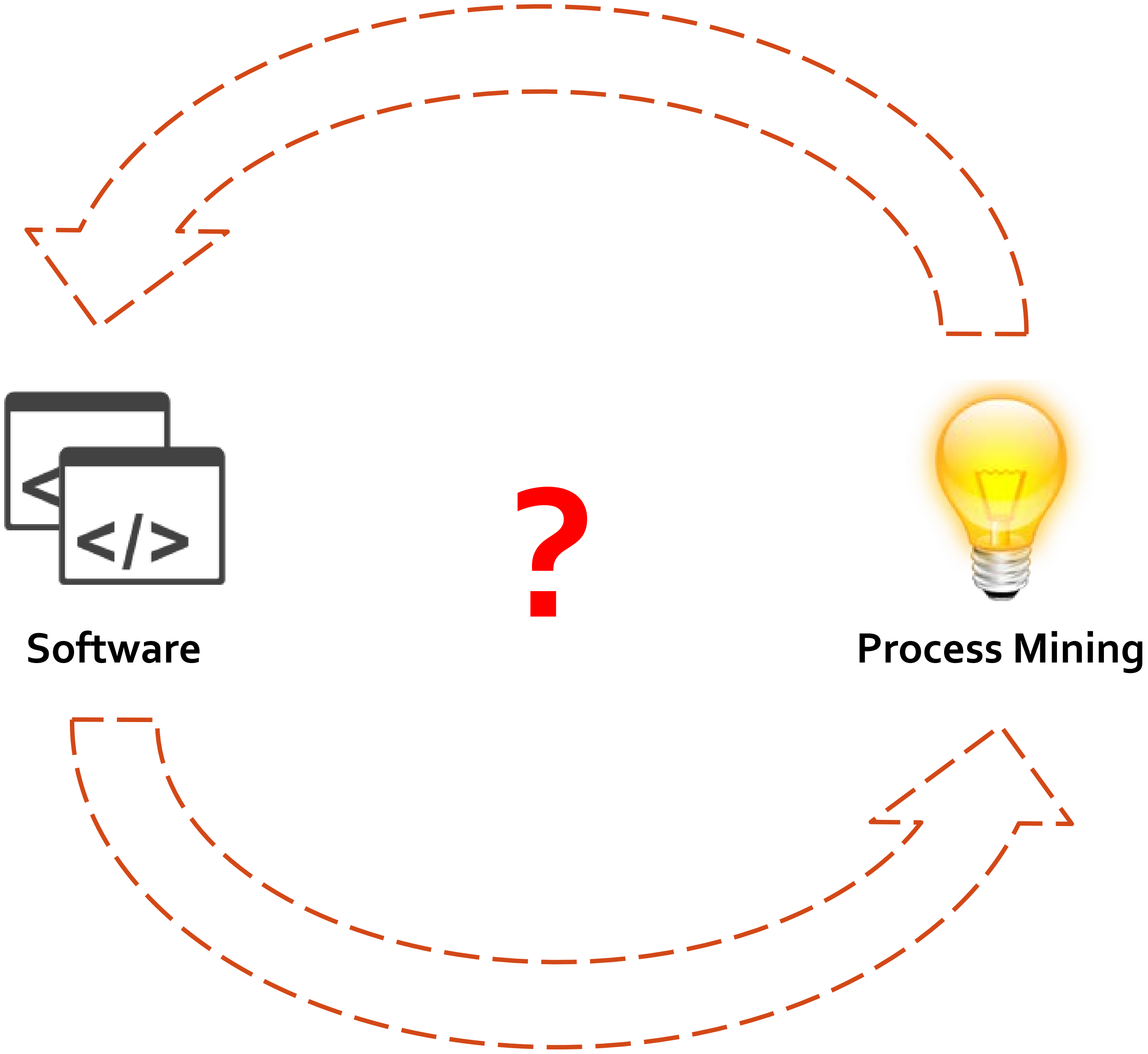
Process Mining for Big Software

Software Instrumentation & Hierarchical Discovery

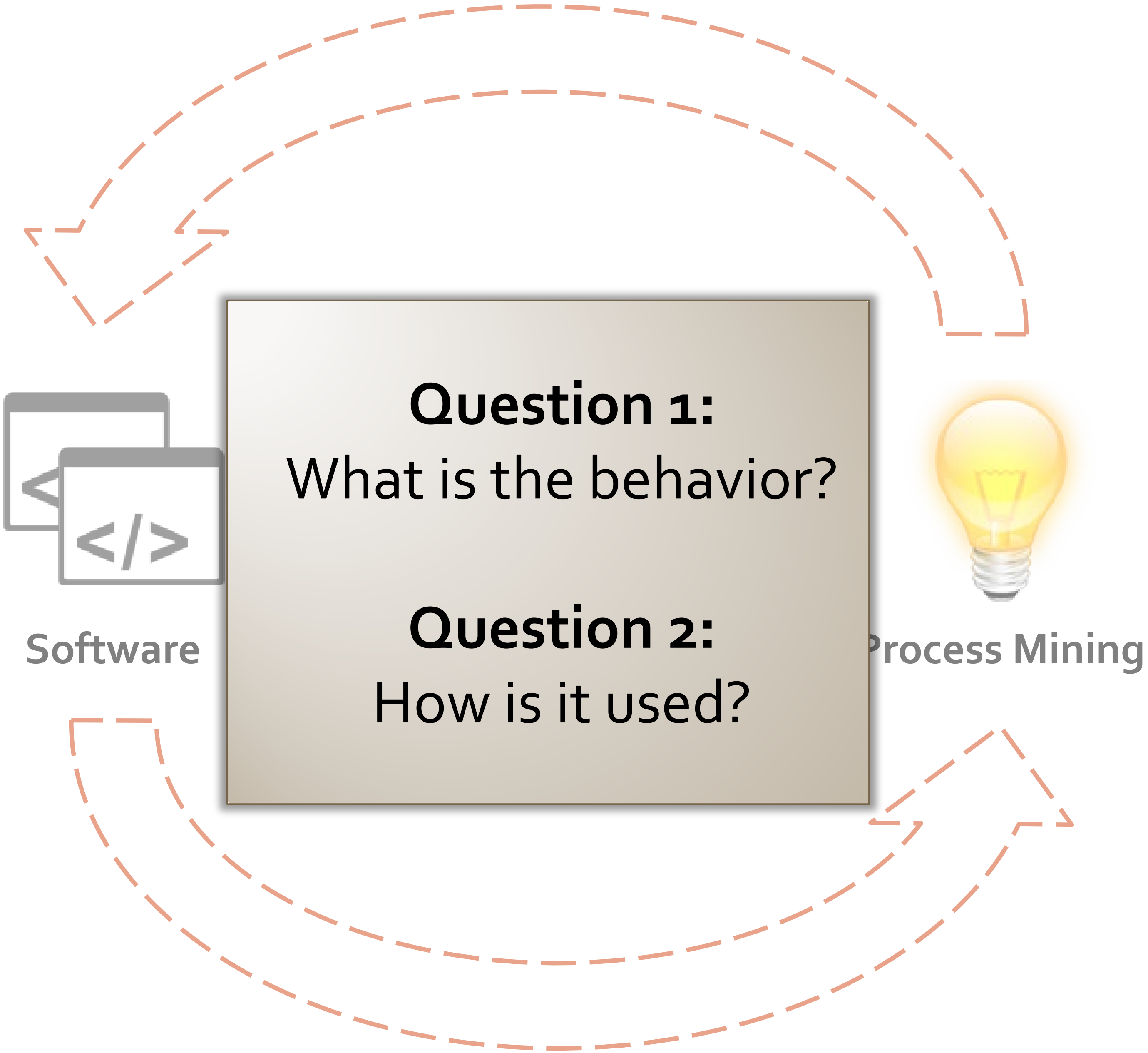
Maikel Leemans (m.leemans@tue.nl)



Software

?

Process Mining



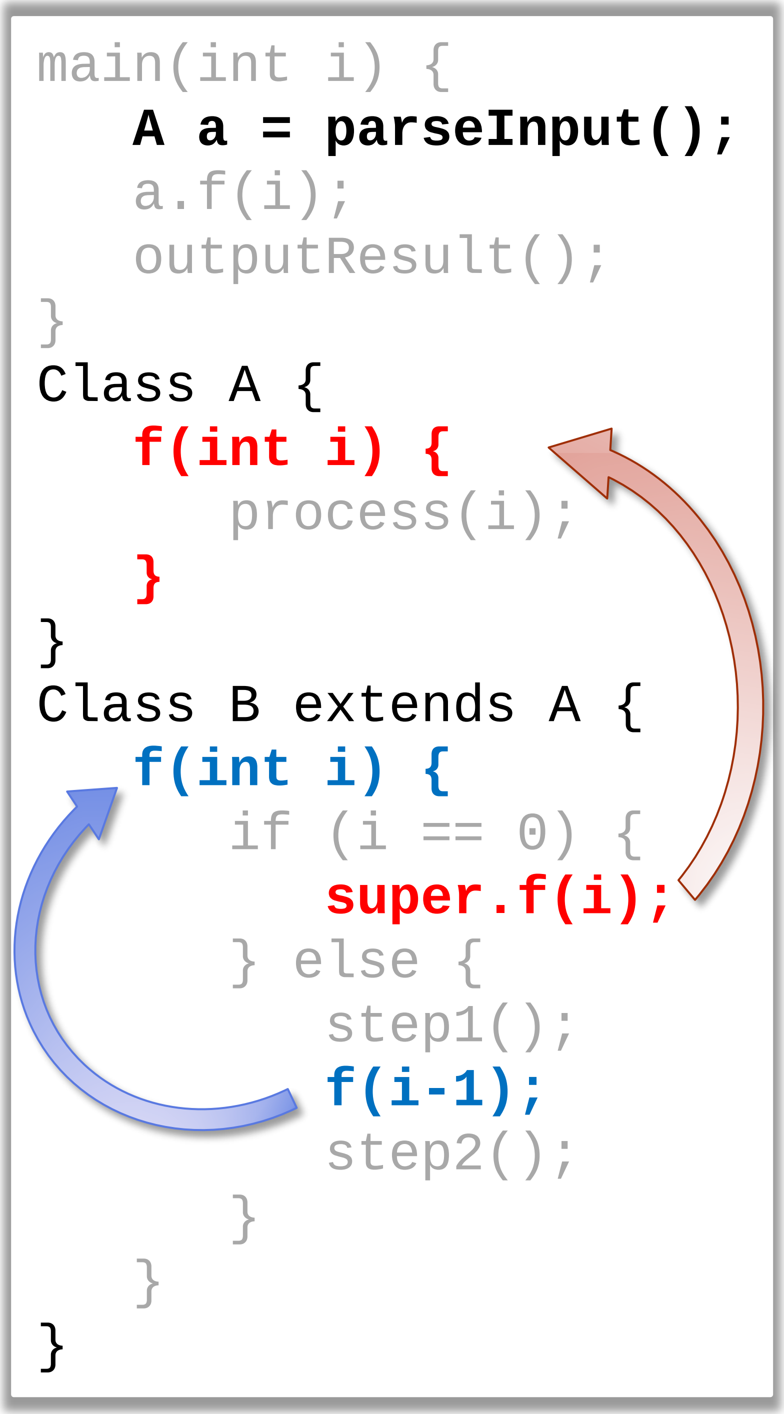
Question 1:
What is the behavior?

Question 2:
How is it used?

```
main(int i) {
   A a = parseInput();
   a.f(i);
   outputResult();
}
```

```
main(int i) {
   A a = parseInput();
   a.f(i);
   outputResult();
}
Class A {
   f(int i) {
      process(i);
   }
}
```

```
main(int i) {
   A a = parseInput();
   a.f(i);
   outputResult();
}
Class A {
   f(int i) {
      process(i);
   }
}
Class B extends A {
   f(int i) {
      if (i == 0) {
         super.f(i);
      } else {
         step1();
         f(i-1);
         step2();
      }
   }
}
```



```
main(int i) {
   A a = parseInput();
   a.f(i);
   outputResult();
}
Class A {
   f(int i) {
      process(i);
   }
}
Class B extends A {
   f(int i) {
      if (i == 0) {
         super.f(i);
      } else {
         step1();
         f(i-1);
         step2();
      }
   }
}
```

yields **A or B ?**

```
main(int i) {
   A a = parseInput();
   a.f(i);
   outputResult();
}
Class A {
   f(int i) {
      process(i);
   }
}
Class B extends A {
   f(int i) {
      if (i == 0) {
         super.f(i);
      } else {
         step1();
         f(i-1);
         step2();
      }
   }
}
```

yields **A or B ?**

Question 1:
What is the behavior?

Question 2:
How is it used?

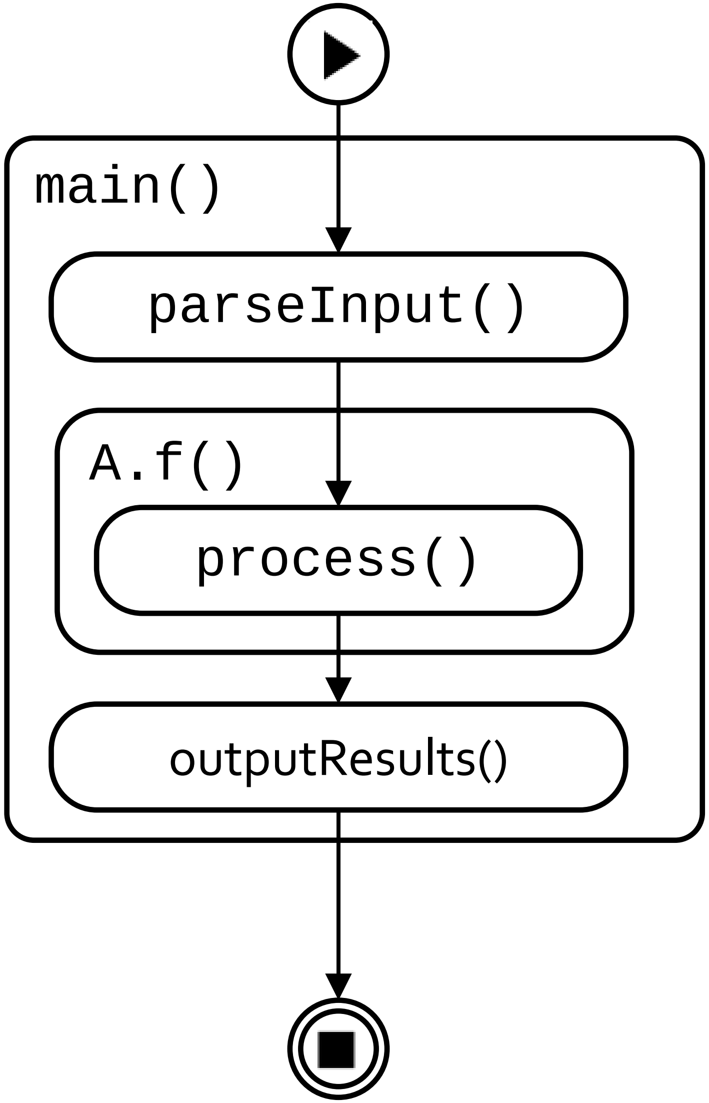
```
main(int i) {
   A a = parseInput();
   a.f(i);
   outputResult();
}
Class A {
   f(int i) {
      process(i);
   }
}
Class B extends A {
   f(int i) {
      if (i == 0) {
         super.f(i);
      } else {
         step1();
         f(i-1);
         step2();
      }
   }
}
```

Question 1:
What is the behavior?

Question 2:
How is it used?

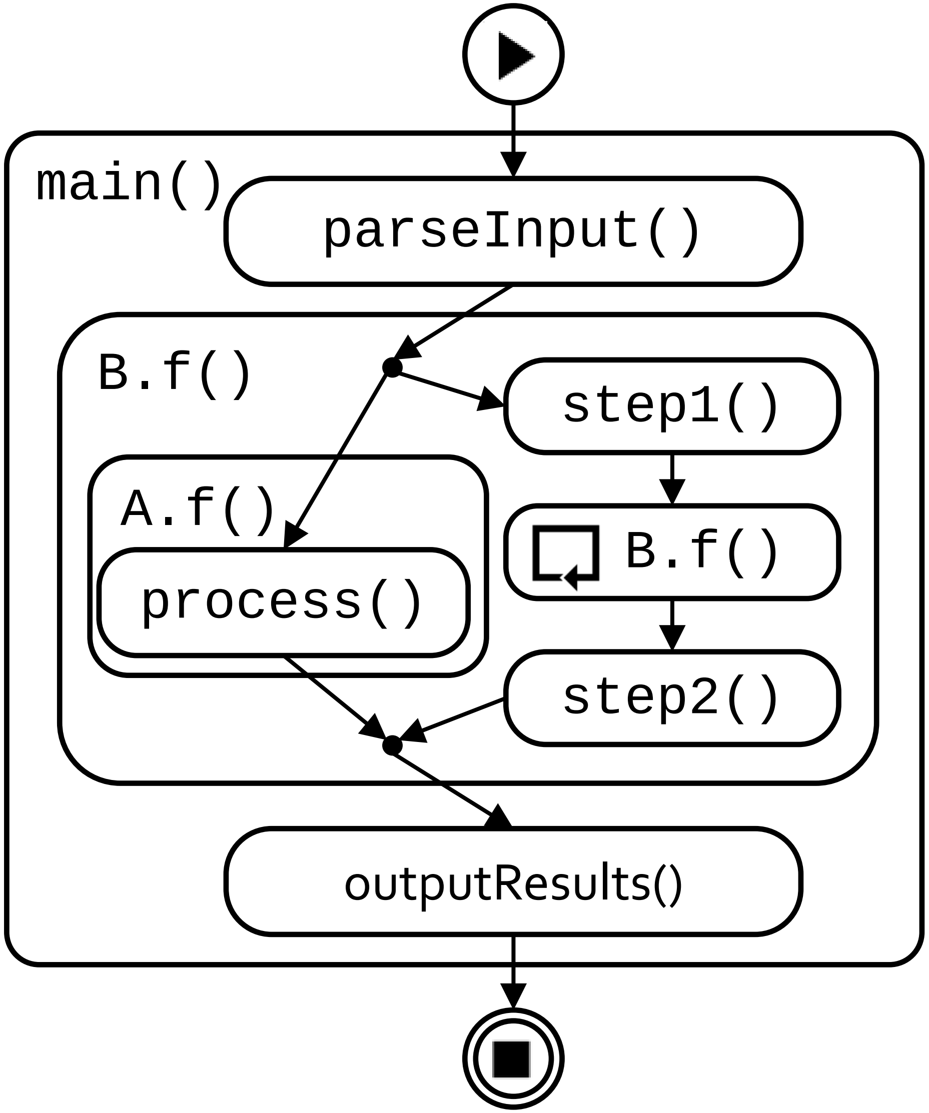
```
main(int i) {
   A a = parseInput();
   a.f(i);
   outputResult();
}
Class A {
   f(int i) {
      process(i);
   }
}
Class B extends A {
   f(int i) {
      if (i == 0) {
         super.f(i);
      } else {
         step1();
         f(i-1);
         step2();
      }
   }
}
```

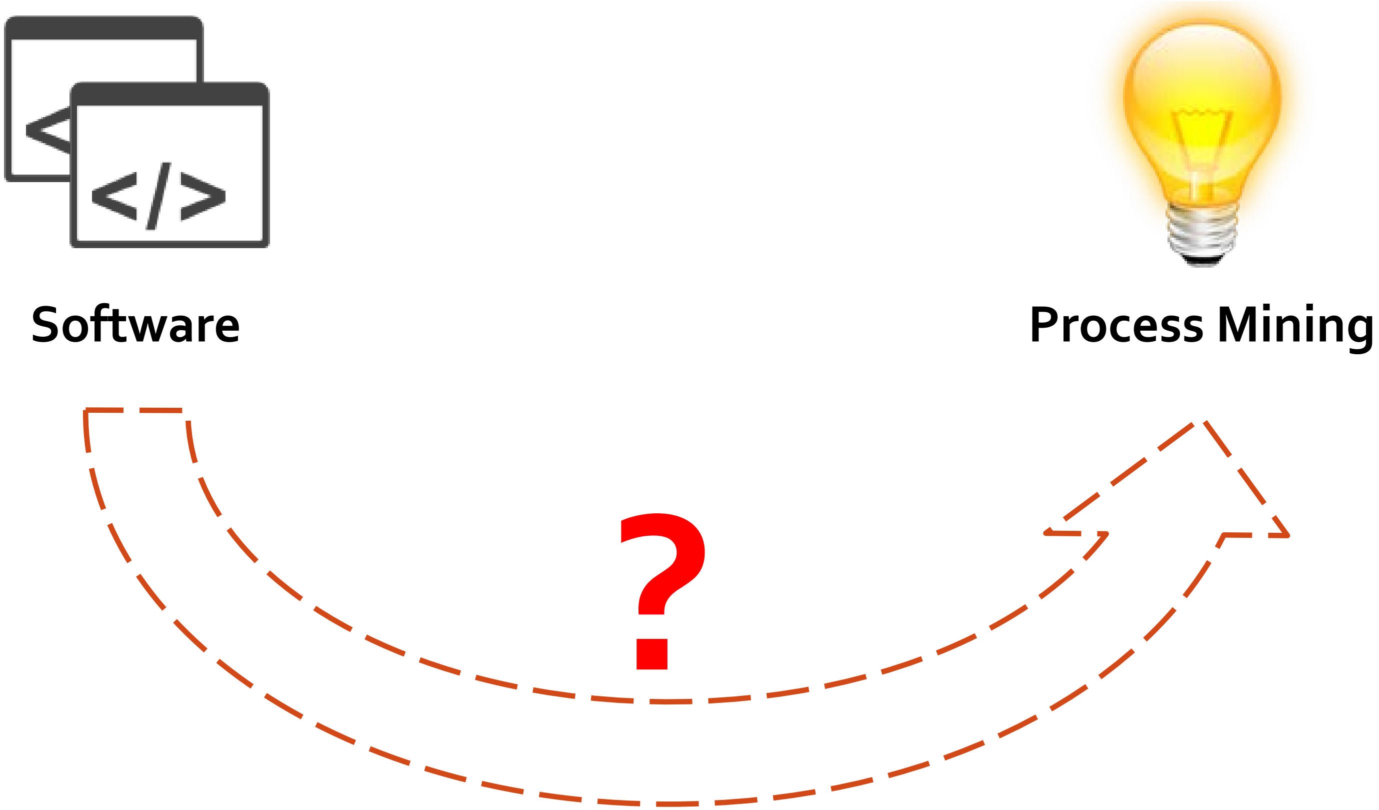
yields **new A**



```
main(int i) {
   A a = parseInput();
   a.f(i);
   outputResult();
}
Class A {
   f(int i) {
      process(i);
   }
}
Class B extends A {
   f(int i) {
      if (i == 0) {
         super.f(i);
      } else {
         step1();
         f(i-1);
         step2();
      }
   }
}
```

yields **new B**

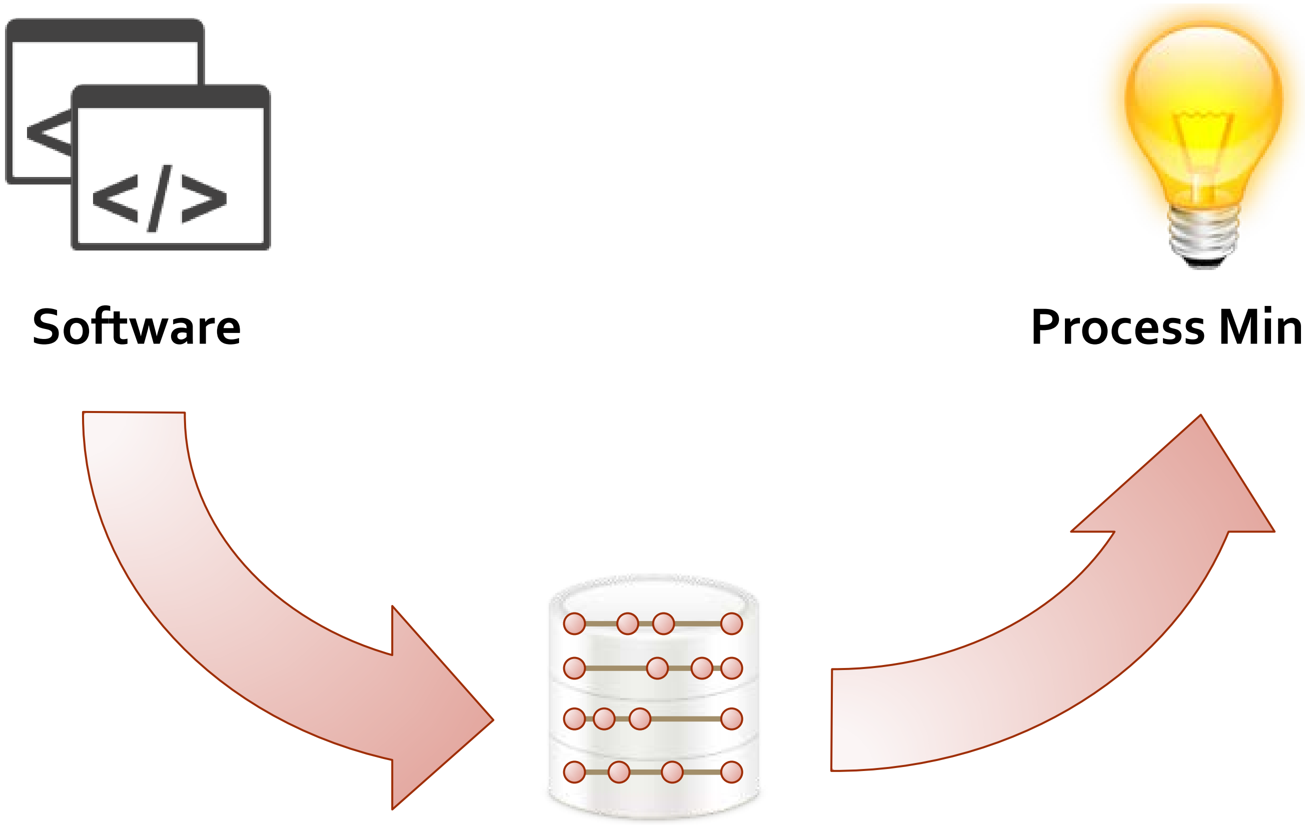




Software

Process Mining

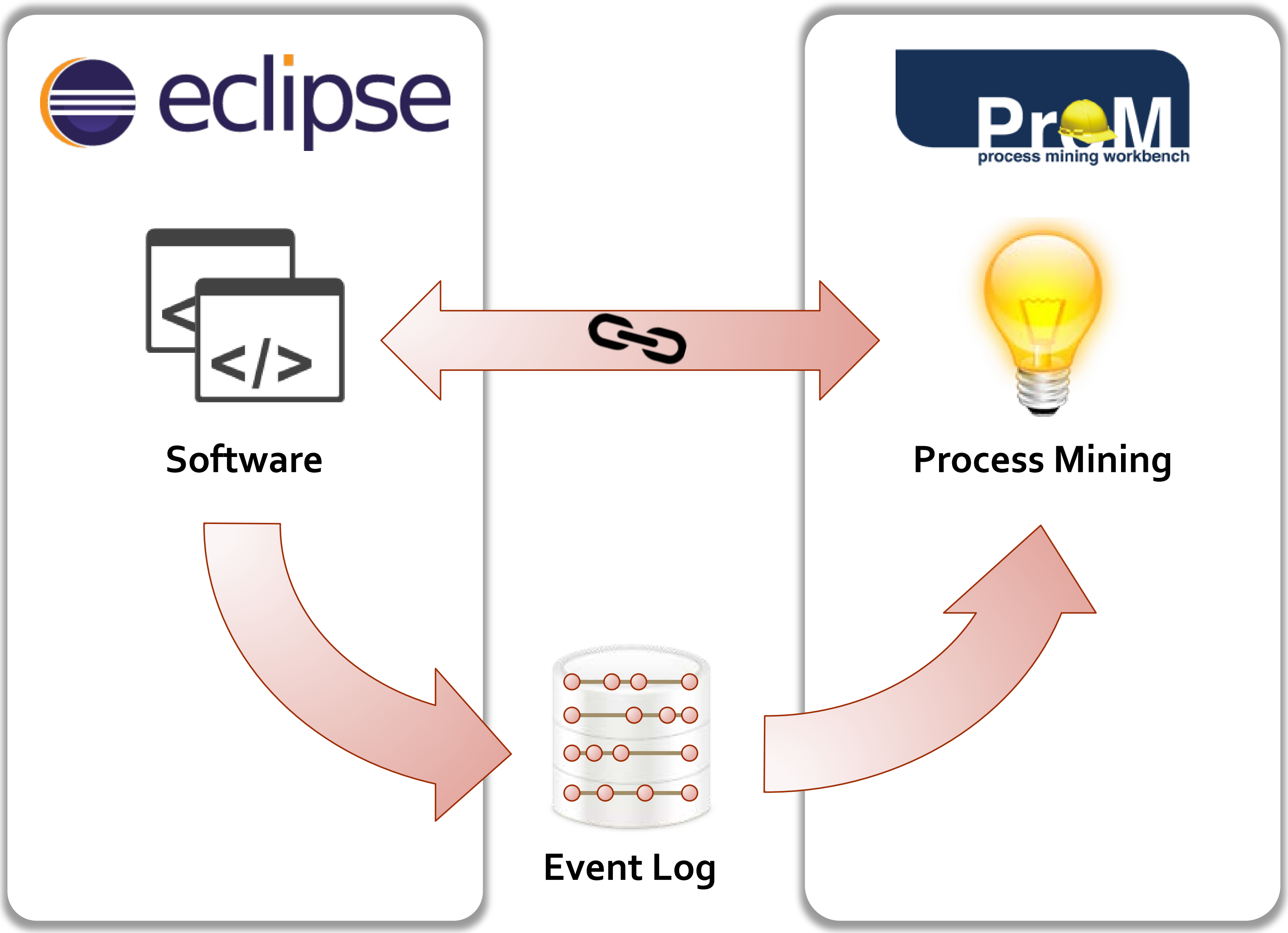
?



Software

Process Mining

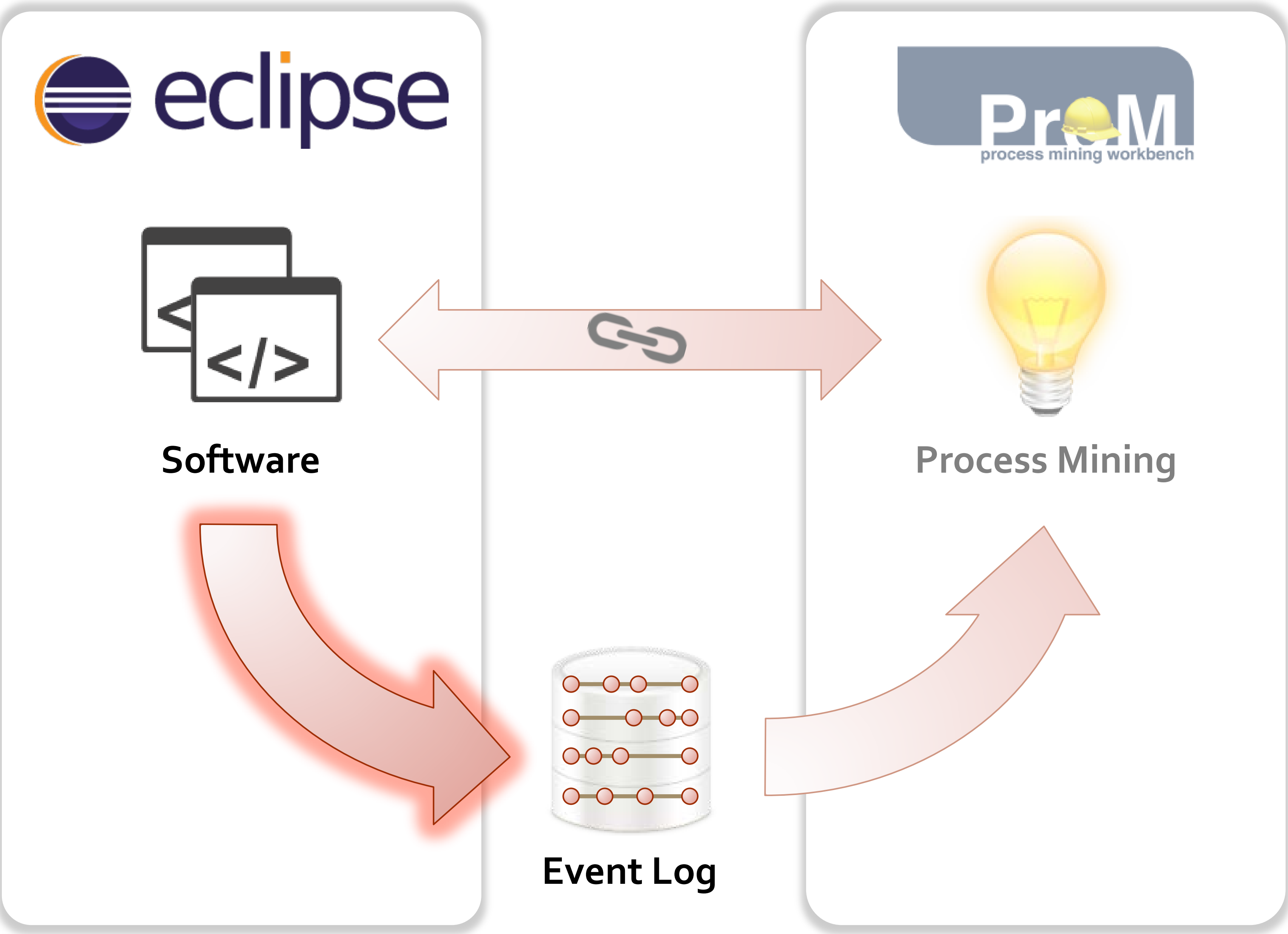
Event Log



Software

Process Mining

Event Log



Software

Process Mining

Event Log

From Software to Event Logs

Software
(Binary)

+

Pointcuts
(Config)

From Software to Event Logs

Software
(Binary)

+

Pointcuts
(Config)

Log method calls

Filter on:

- Packages
- Classes
- Interfaces
- Method names

From Software to Event L

Software
(Binary)

Pointcuts
(Config)

```
main(int i) {
   A a = parseInput();
   a.f(i);
   outputResult();
}
Class A {
   f(int i) {
      process(i);
   }
}
Class B extends A {
   f(int i) {
      if (i == 0) {
         super.f(i);
      } else {
         step1();
         f(i-1);
         step2();
      }
   }
}
```

From Software to Event L

Software
(Binary)

+

Pointcuts
(Config)

Log: call main()

Log: return main()

Log: call A.f()

Log: return A.f()

Log: call B.f()

Log: return B.f()

```
main(int i) {
   A a = parseInput();
   a.f(i);
   outputResult();
}
Class A {
   f(int i) {
      process(i);
   }
}
Class B extends A {
   f(int i) {
      if (i == 0) {
         super.f(i);
      } else {
         step1();
         f(i-1);
         step2();
      }
   }
}
```

From Software to Event L

Log: call main()

Log: return main()

Log:

Log: return B.f()

```
main(int i) {
    A a = parseInput();
    a.f(i);
    outputResult();
}
Class A {
    f(int i) {
```

```
            step1();
            f(i-1);
            step2();
        }
    }
}
```

Software (Binary)

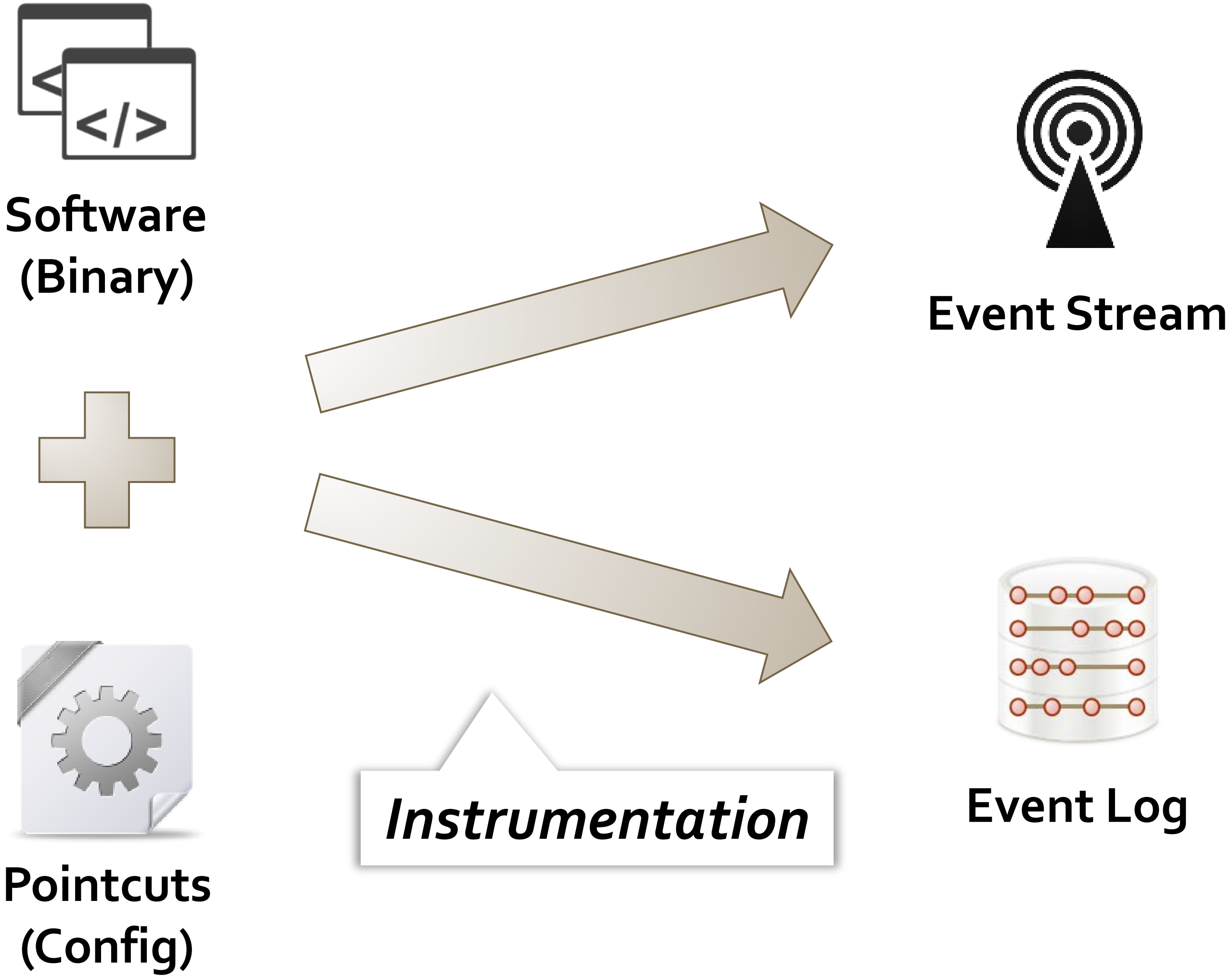
+

Pointcuts (Config)

Logged Event:

Case	Software Run 1
Activity	org.package.ClassA.main(...) + return
Time	04-10-2016 T 12:00:00.126
Resource	thread-1
Source	ClassA.java @ line 25

From Software to Event Logs



Software (Binary)

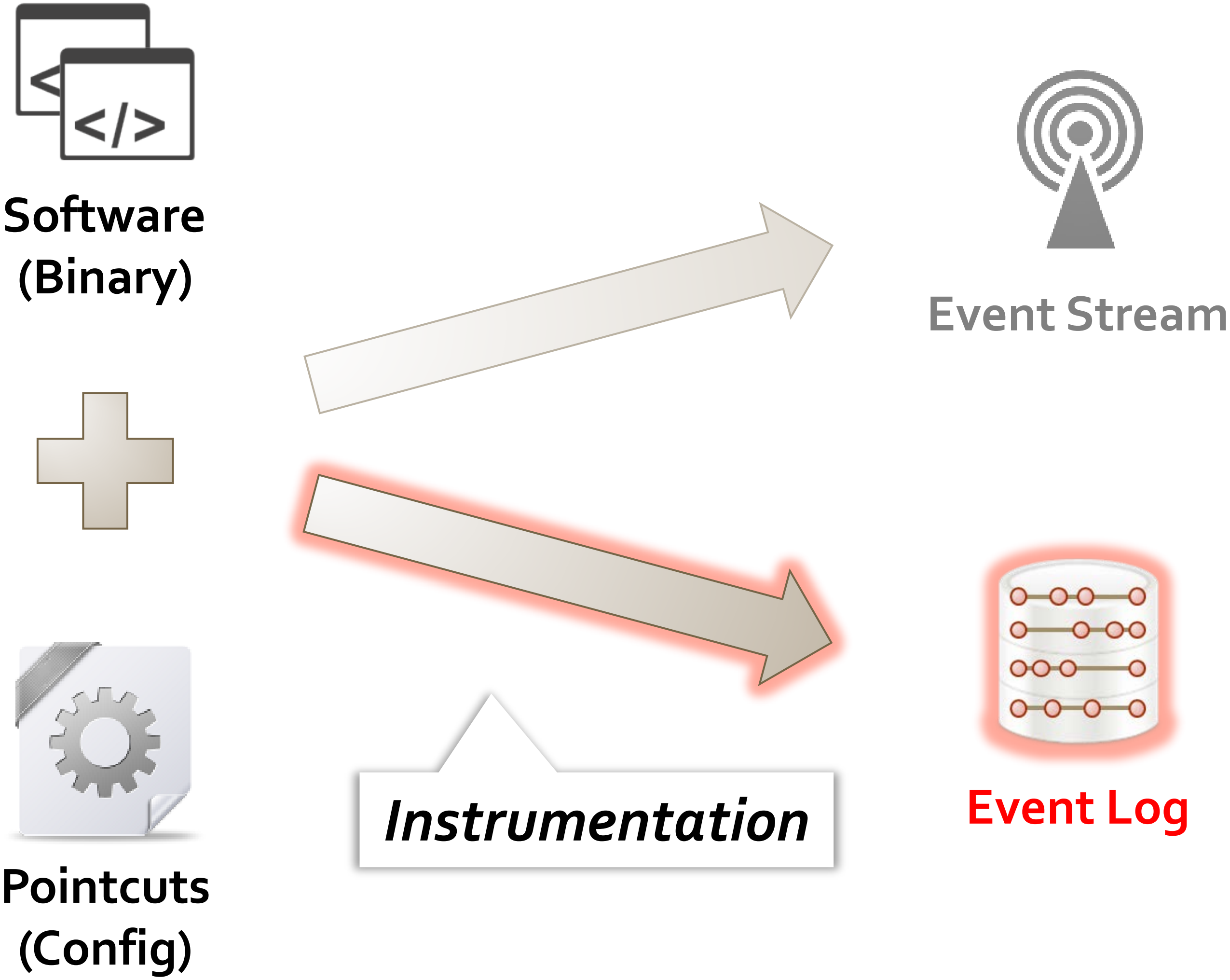
Pointcuts (Config)

Instrumentation

Event Stream

Event Log

From Software to Event Logs



Software (Binary)

Pointcuts (Config)

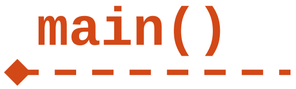
Instrumentation

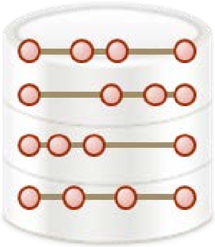
Event Stream

Event Log

From Software to an Event Log

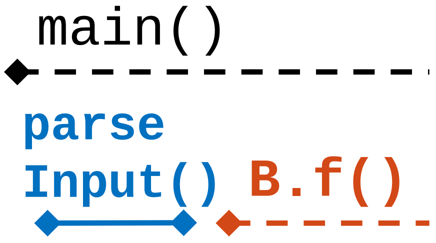
```
main(int i) {
   A a = parseInput();
   a.f(i);
   outputResult();
}
Class A {
   f(int i) {
      process(i);
   }
}
Class B extends A {
   f(int i) {
      if (i == 0) {
         super.f(i);
      } else {
         step1();
         f(i-1);
         step2();
      }
   }
}
```

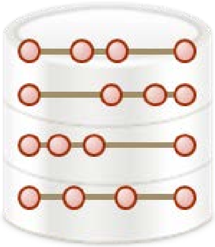




```
main(int i) {
   A a = parseInput();
   a.f(i);
   outputResult();
}
Class A {
   f(int i) {
      process(i);
   }
}
Class B extends A {
   f(int i) {
      if (i == 0) {
         super.f(i);
      } else {
         step1();
         f(i-1);
         step2();
      }
   }
}
```

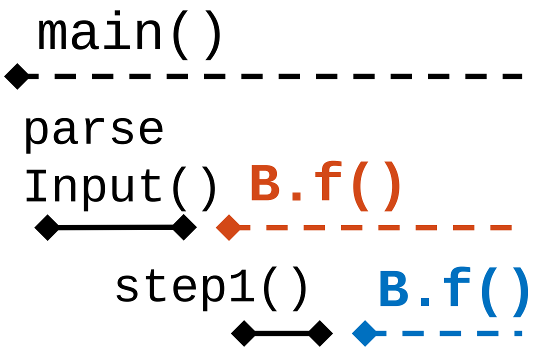
From Software to an Event Log

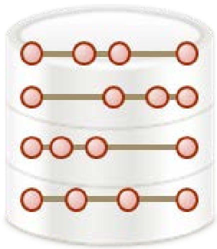




```
main(int i) {
   A a = parseInput();
   a.f(i);
   outputResult();
}
Class A {
   f(int i) {
      process(i);
   }
}
Class B extends A {
   f(int i) {
      if (i == 0) {
         super.f(i);
      } else {
         step1();
         f(i-1);
         step2();
      }
   }
}
```

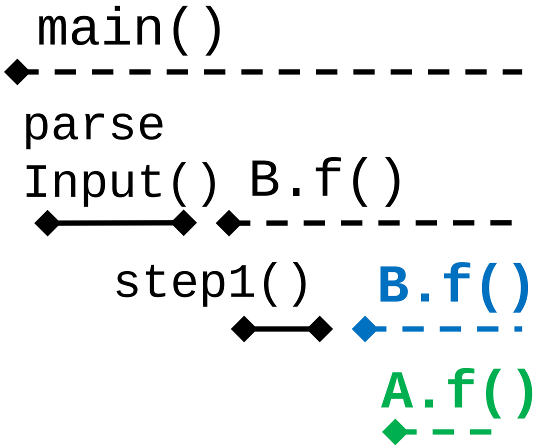
From Software to an Event Log

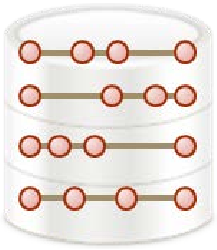




From Software to an Event Log

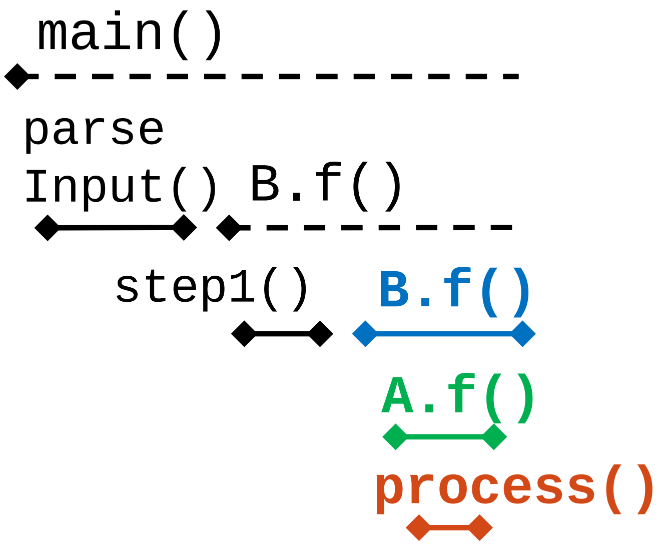
```
main(int i) {
   A a = parseInput();
   a.f(i);
   outputResult();
}
Class A {
   f(int i) {
      process(i);
   }
}
Class B extends A {
   f(int i) {
      if (i == 0) {
         super.f(i);
      } else {
         step1();
         f(i-1);
         step2();
      }
   }
}
```

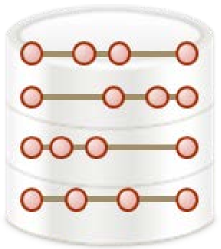




From Software to an Event Log

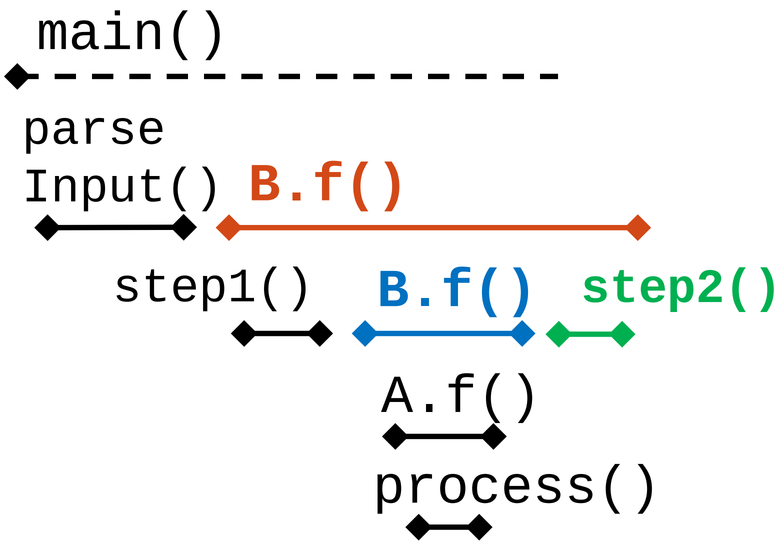
```
main(int i) {
   A a = parseInput();
   a.f(i);
   outputResult();
}
Class A {
   f(int i) {
      process(i);
   }
}
Class B extends A {
   f(int i) {
      if (i == 0) {
         super.f(i);
      } else {
         step1();
         f(i-1);
         step2();
      }
   }
}
```





```
main(int i) {
    A a = parseInput();
    a.f(i);
    outputResult();
}
Class A {
    f(int i) {
        process(i);
    }
}
Class B extends A {
    f(int i) {
        if (i == 0) {
            super.f(i);
        } else {
            step1();
            f(i-1);
            step2();
        }
    }
}
```

From Software to an Event Log

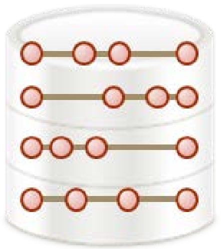

main()

parse Input() **B.f()**

step1() **B.f()** **step2()**

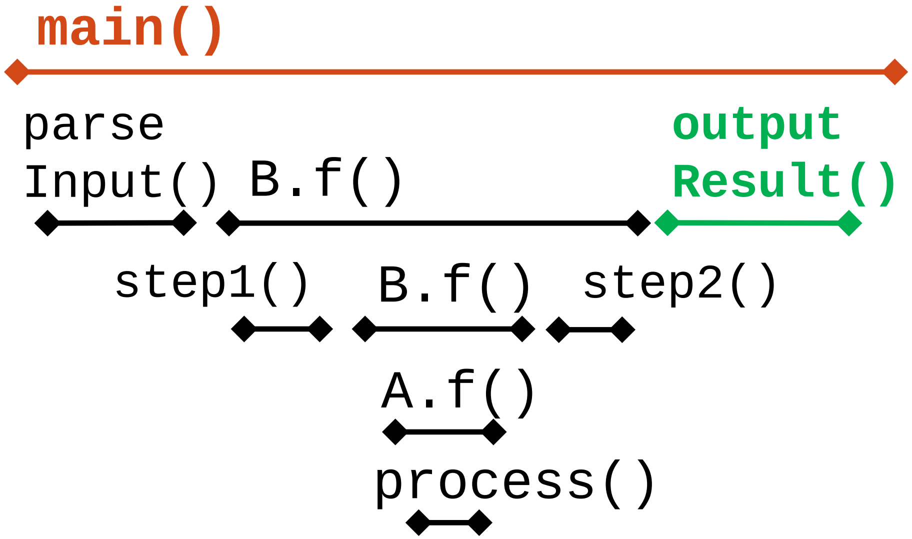
A.f()

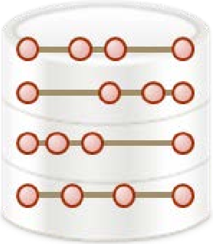
process()



```
main(int i) {
   A a = parseInput();
   a.f(i);
   outputResult();
}
Class A {
   f(int i) {
      process(i);
   }
}
Class B extends A {
   f(int i) {
      if (i == 0) {
         super.f(i);
      } else {
         step1();
         f(i-1);
         step2();
      }
   }
}
```

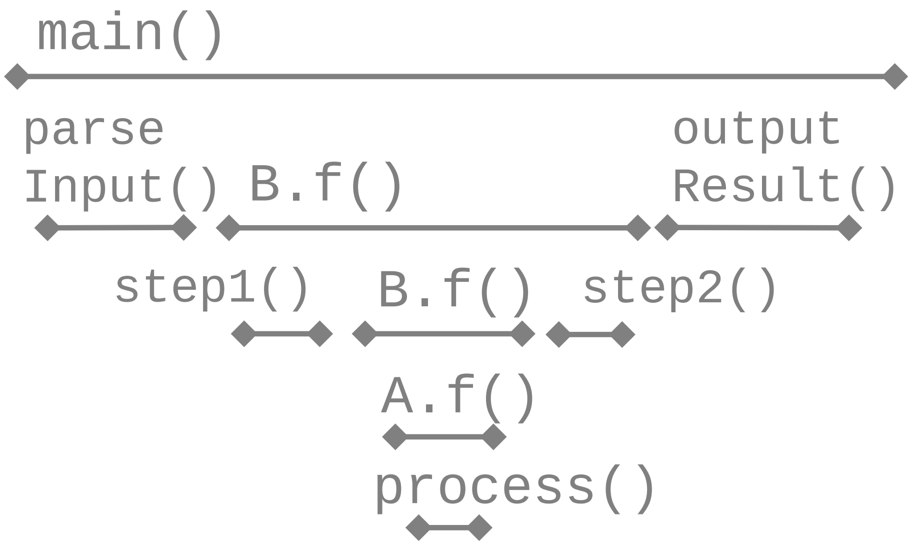
From Software to an Event Log

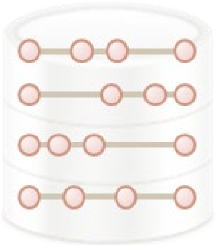


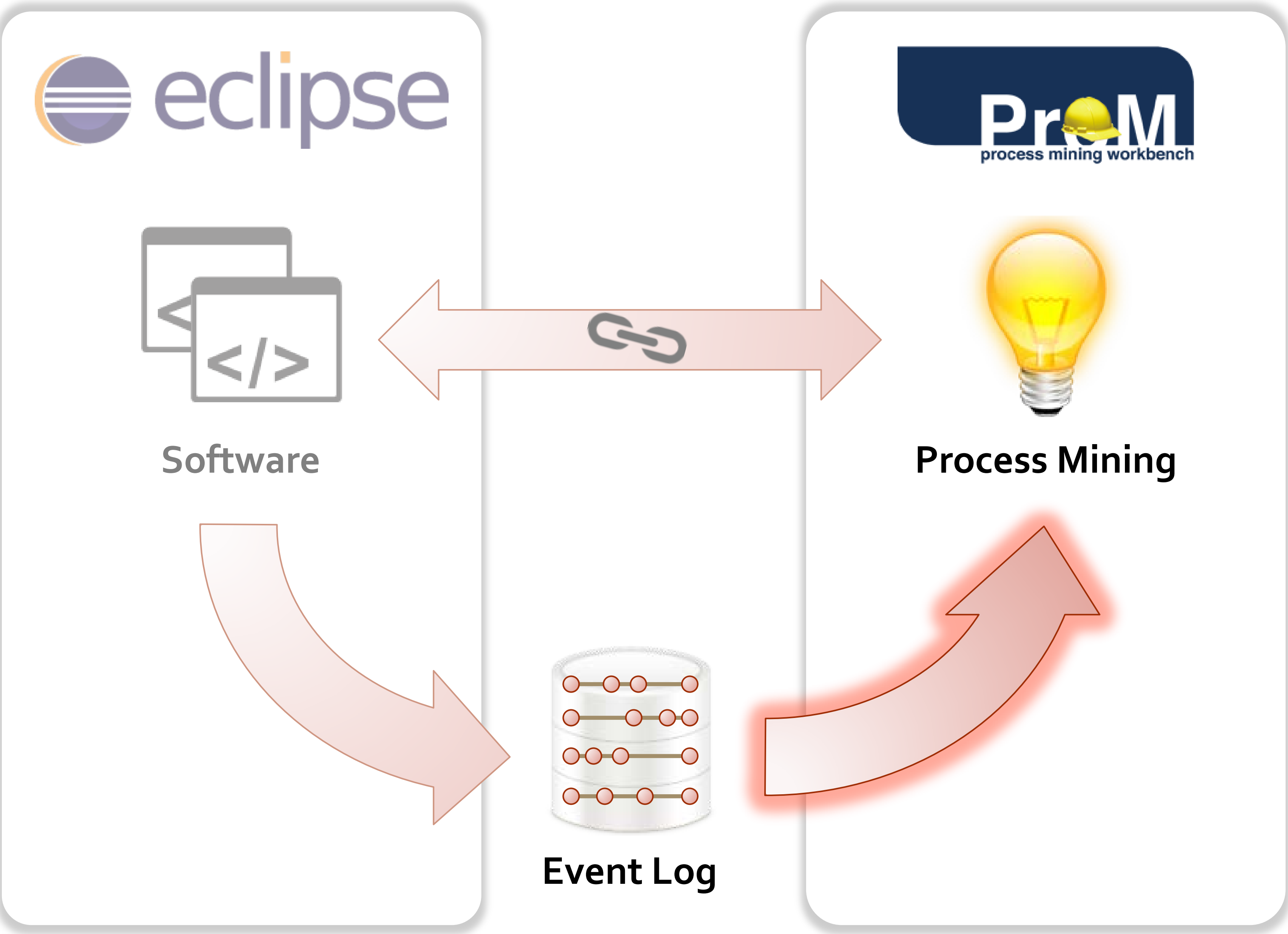


```
main(int i) {
   A a = parseInp
   a.f(i);
   outputResult()
}
Class A {
   f(int i) {
      process(i);
   }
}
Class B extends A {
   f(int i) {
      if (i == 0) {
         super.f(i);
      } else {
         step1();
         f(i-1);
         step2();
      }
   }
}
```

https://svn.win.tue.nl/repos/prom/XPort/





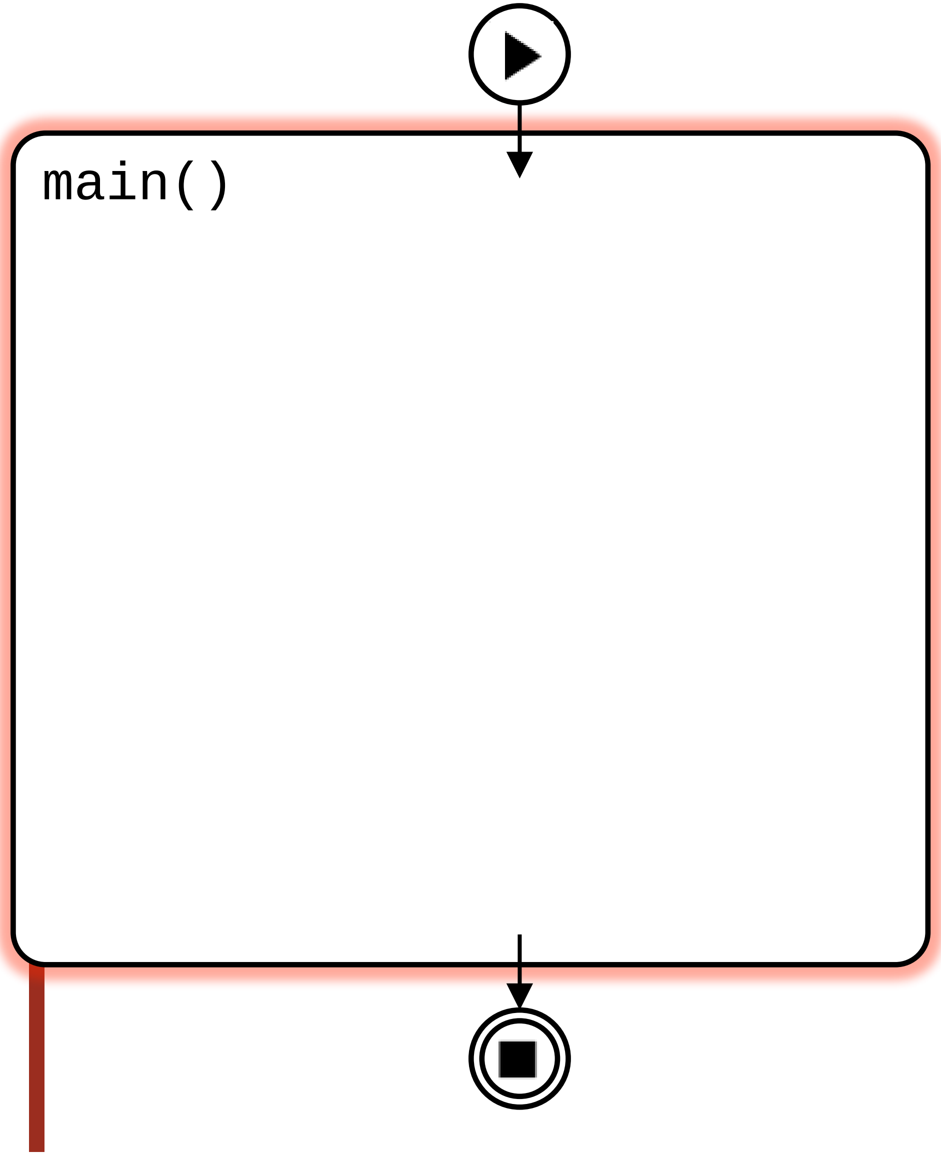


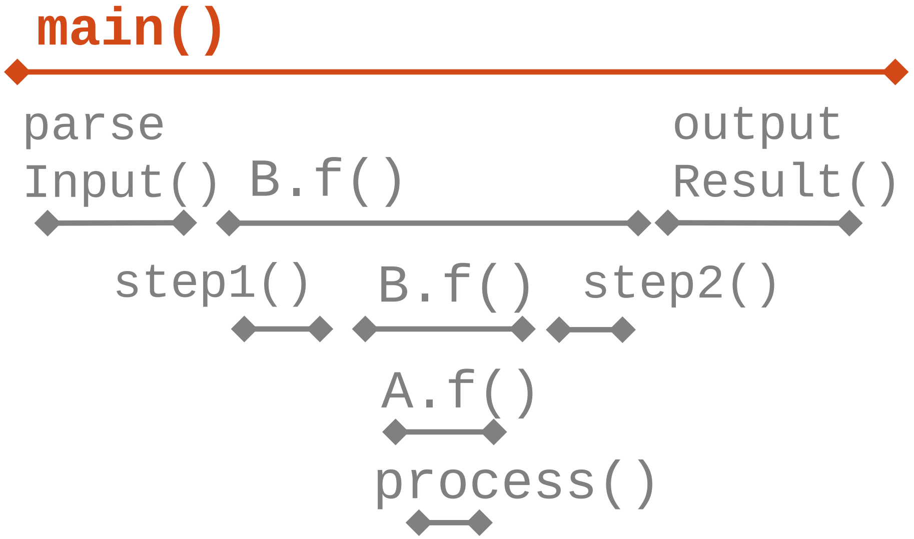
Software

Process Mining

Event Log

Hierarchical Discovery





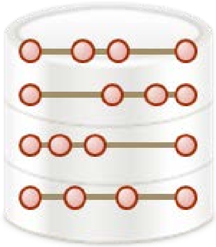
main()

parseInput() B.f() outputResult()

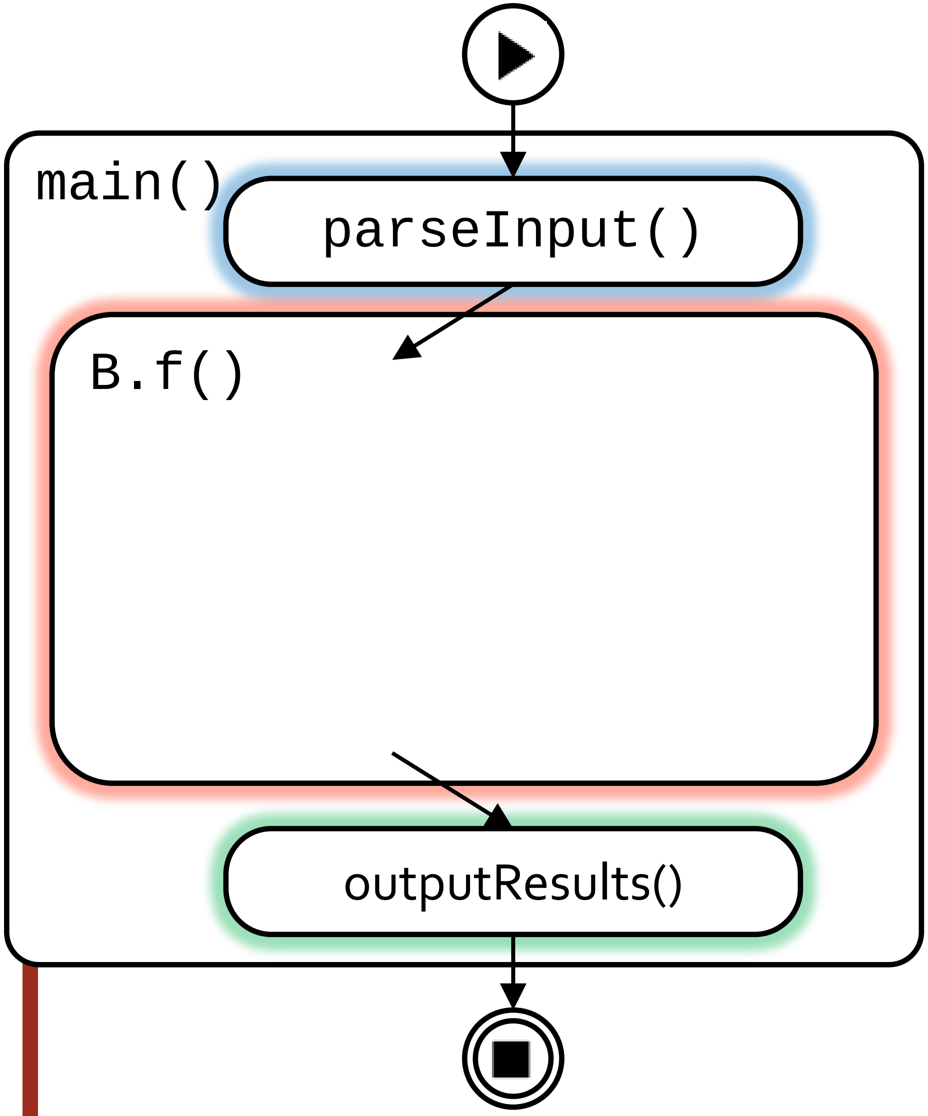
step1() B.f() step2()

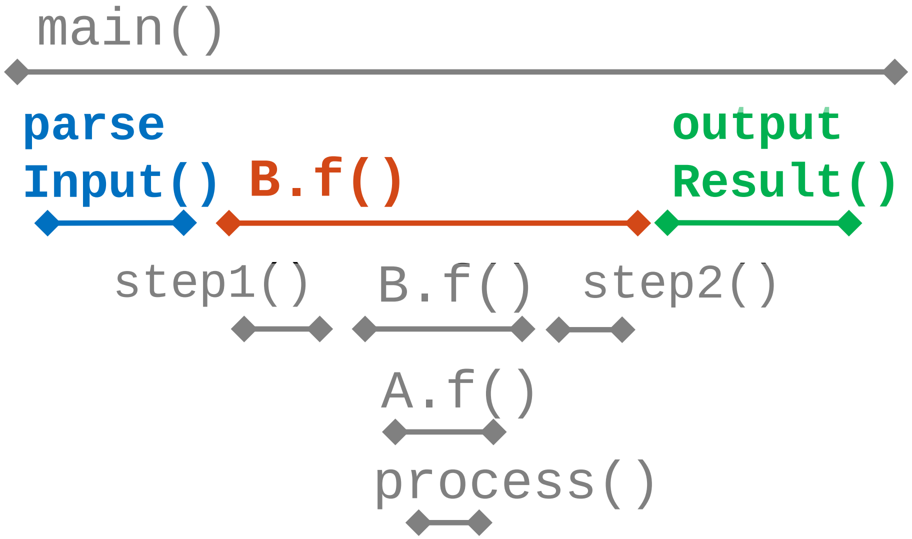
A.f()

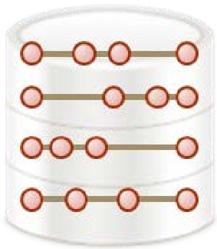
process()



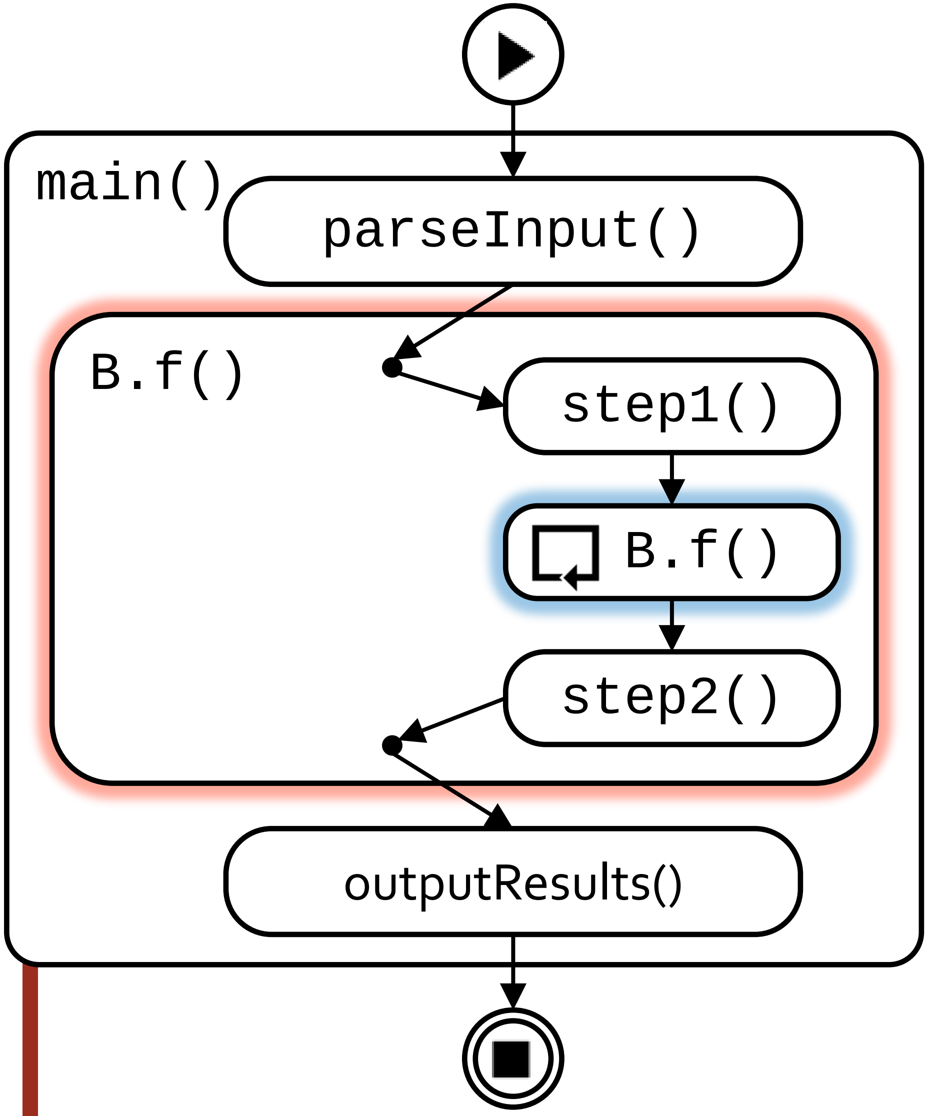
Hierarchical Discovery

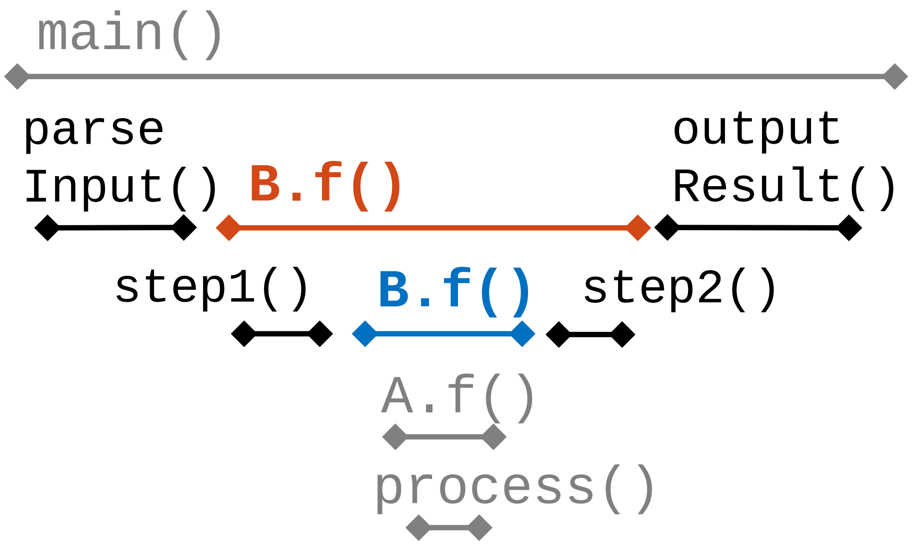


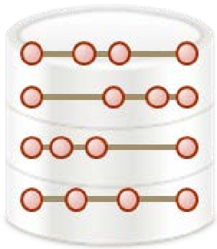




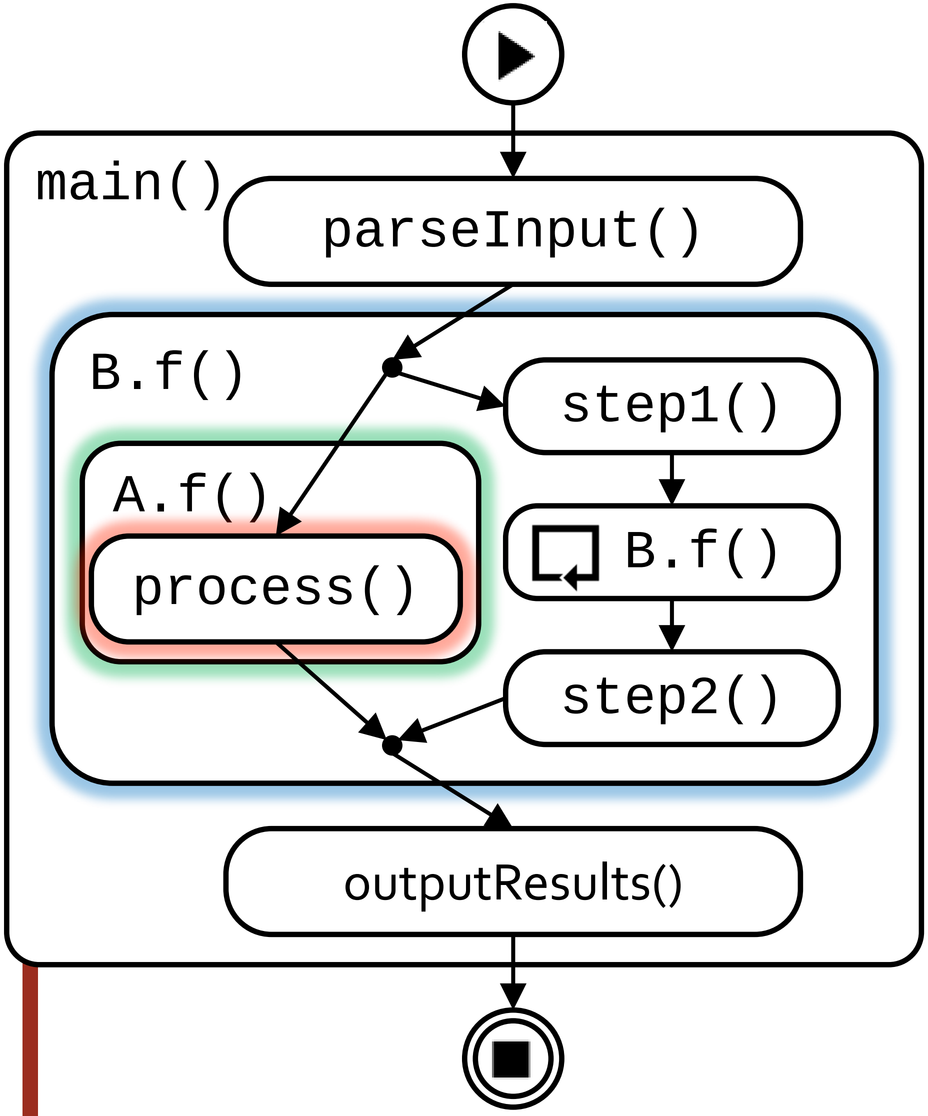
Hierarchical Discovery

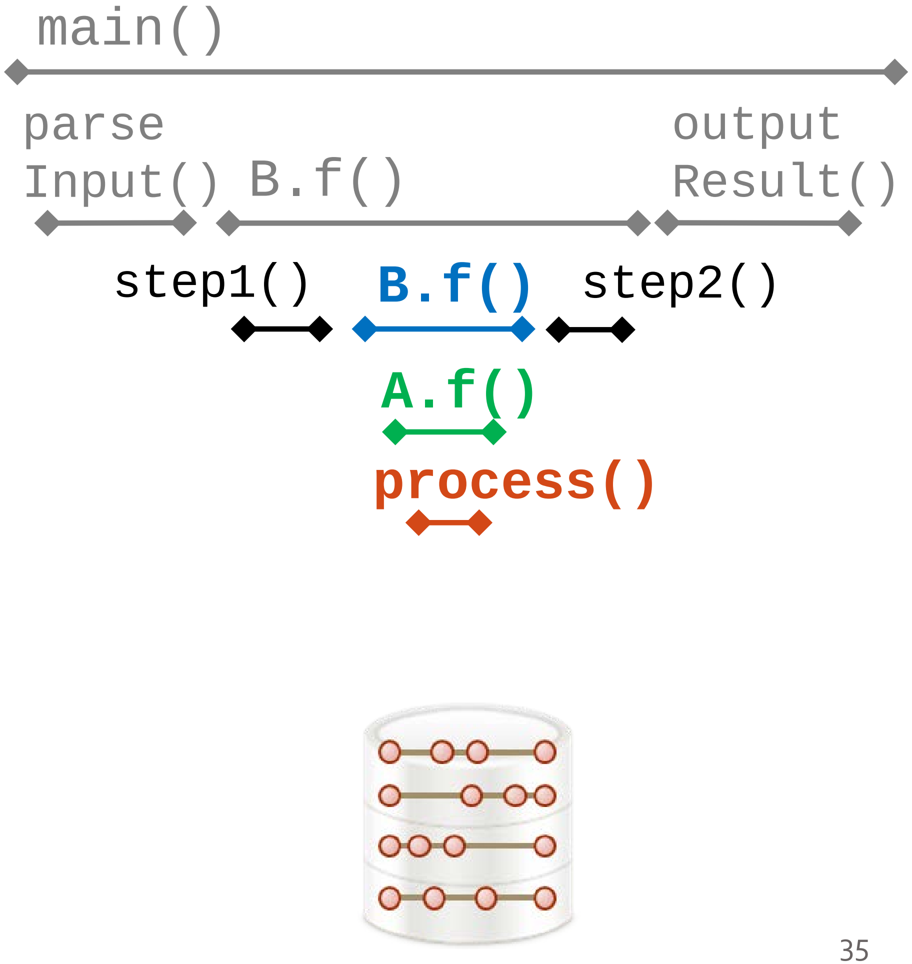






Hierarchical Discovery

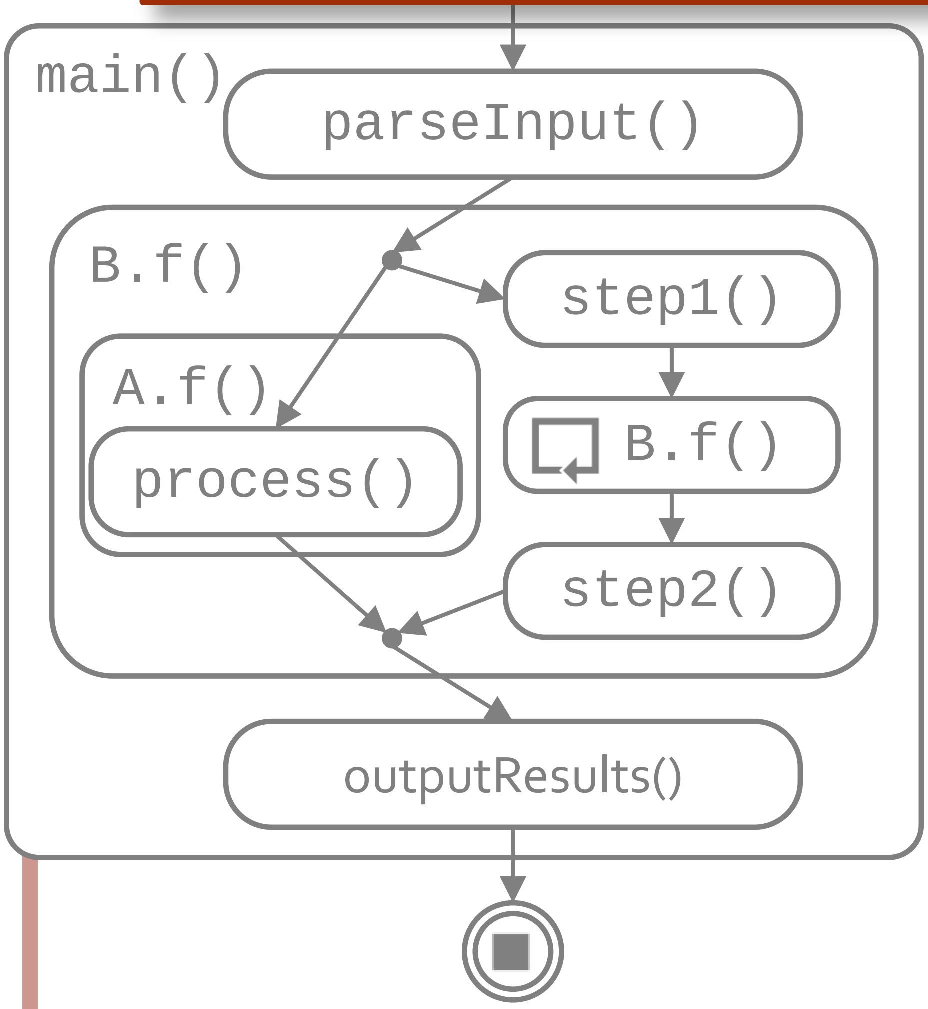


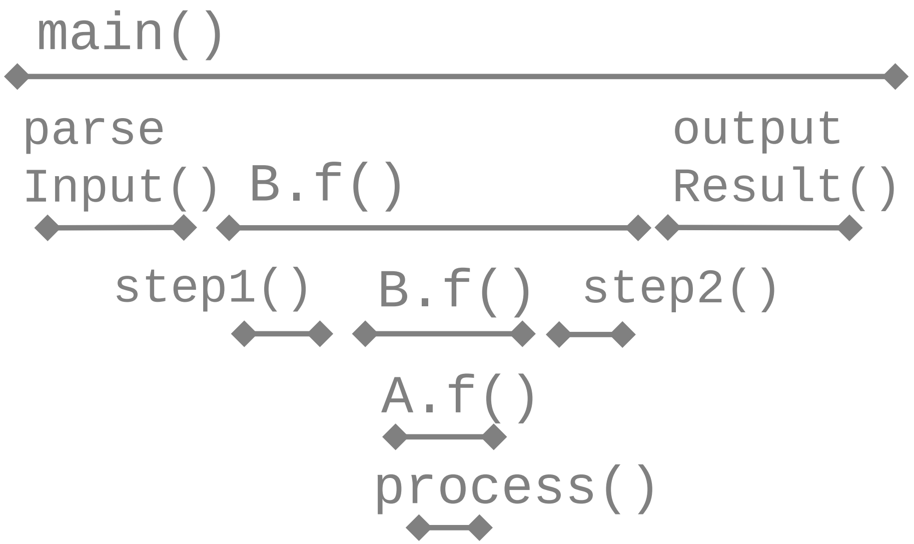


H

ProM
process mining workbench

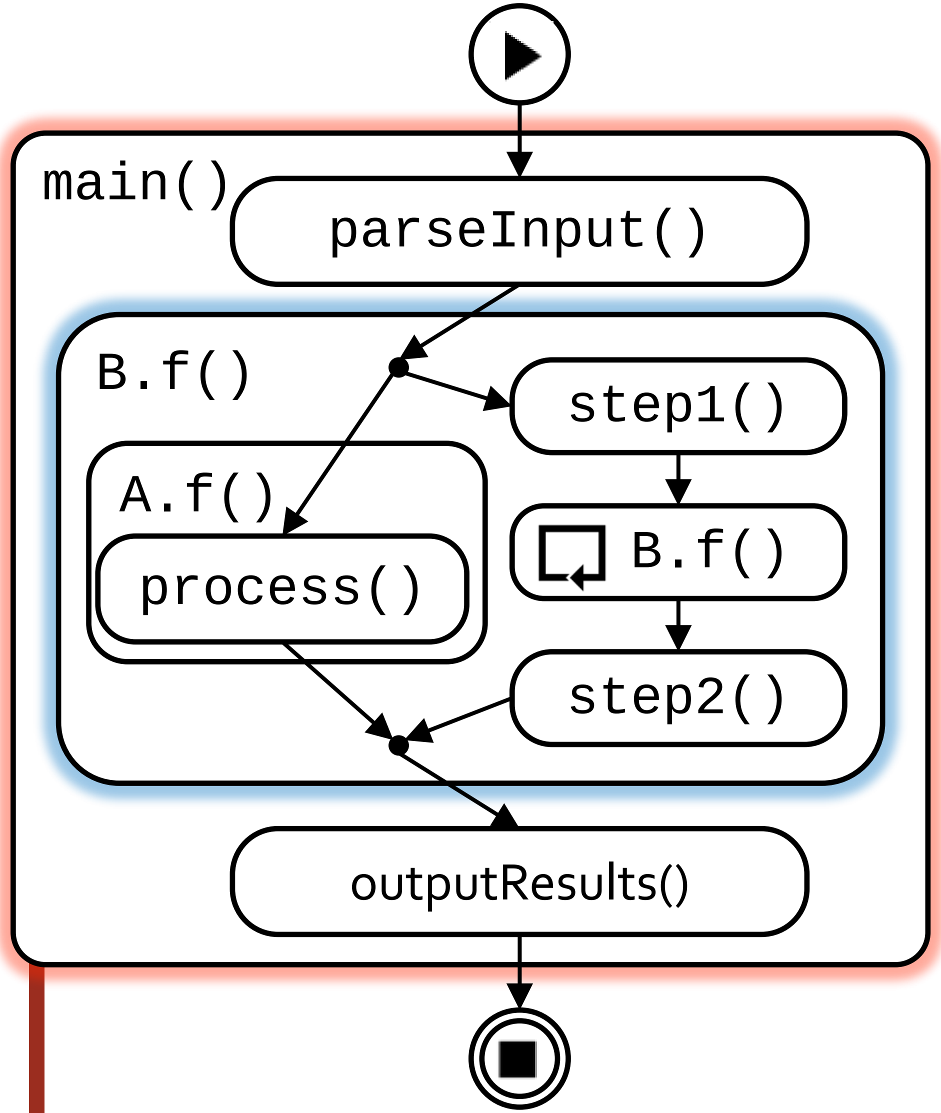
https://svn.win.tue.nl/repos/prom/Packages/Statechart/



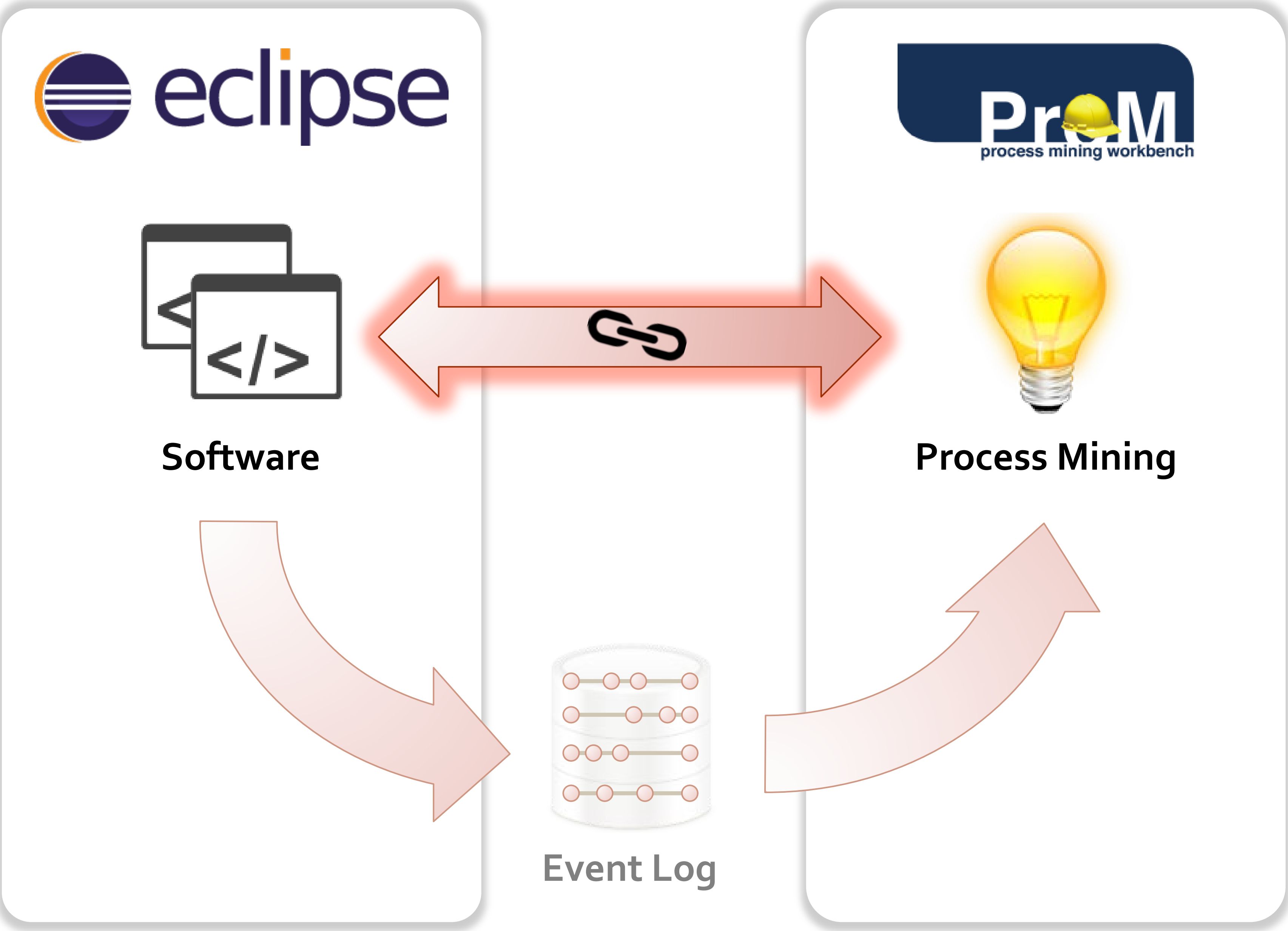


Hierarchical Discovery

yielded **new B**



```
main(int i) {
    A a = parseInput();
    a.f(i);
    outputResult();
}
Class A {
    f(int i) {
        process(i);
    }
}
Class B extends A {
    f(int i) {
        if (i == 0) {
            super.f(i);
        } else {
            step1();
            f(i-1);
            step2();
        }
    }
}
```

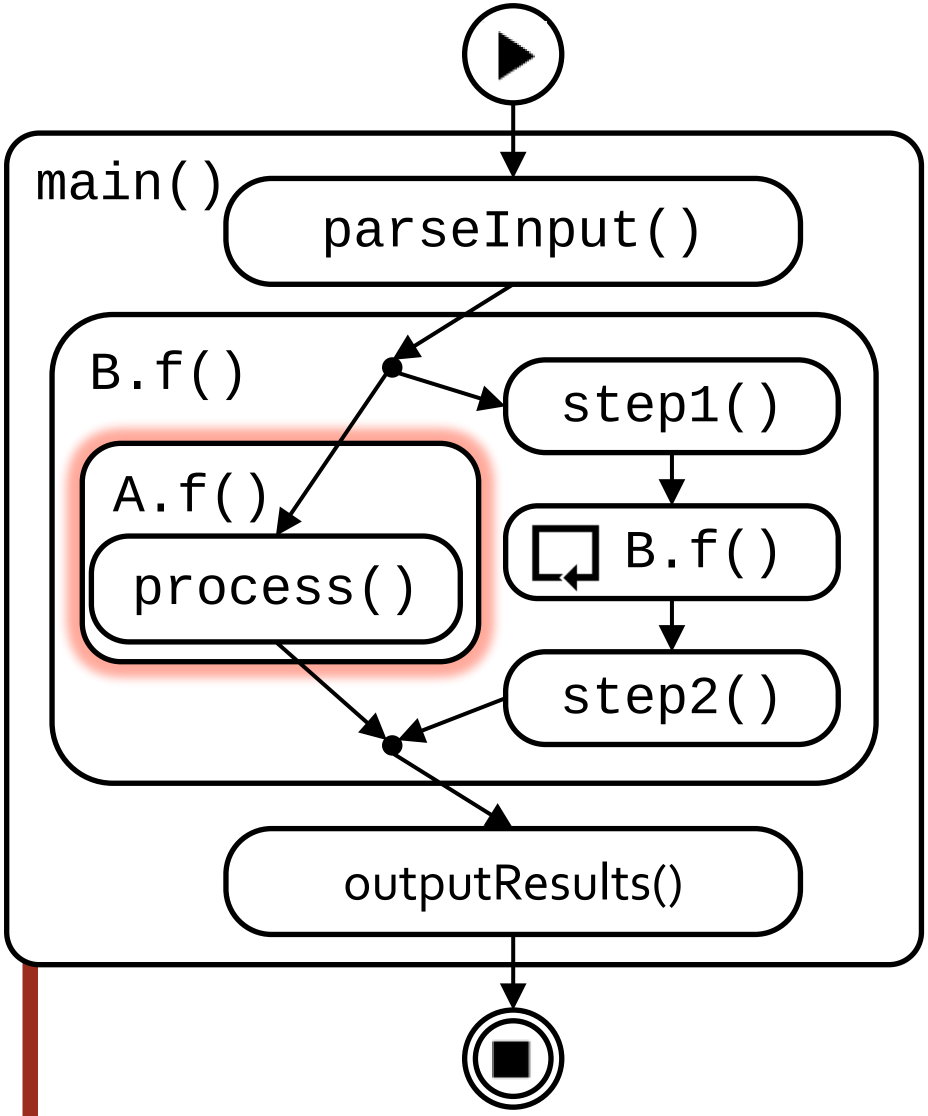


Software

Process Mining

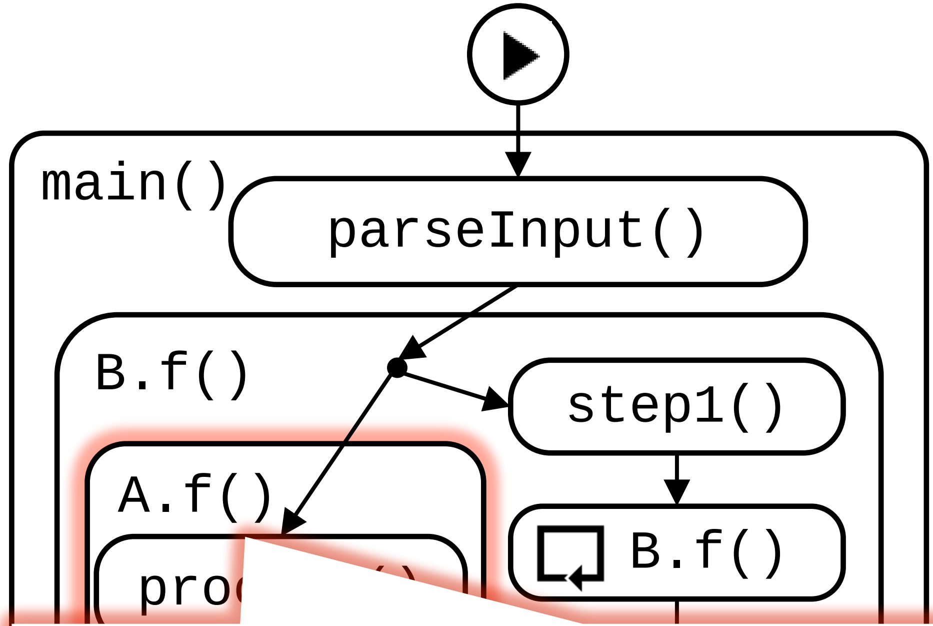
Event Log

Bridge between model and code



```
main(int i) {
   A a = parseInput();
   a.f(i);
   outputResult();
}
Class A {
   f(int i) {
      process(i);
   }
}
Class B extends A {
   f(int i) {
      if (i == 0) {
         super.f(i);
      } else {
         step1();
         f(i-1);
         step2();
      }
   }
}
```

Bridge between model and code

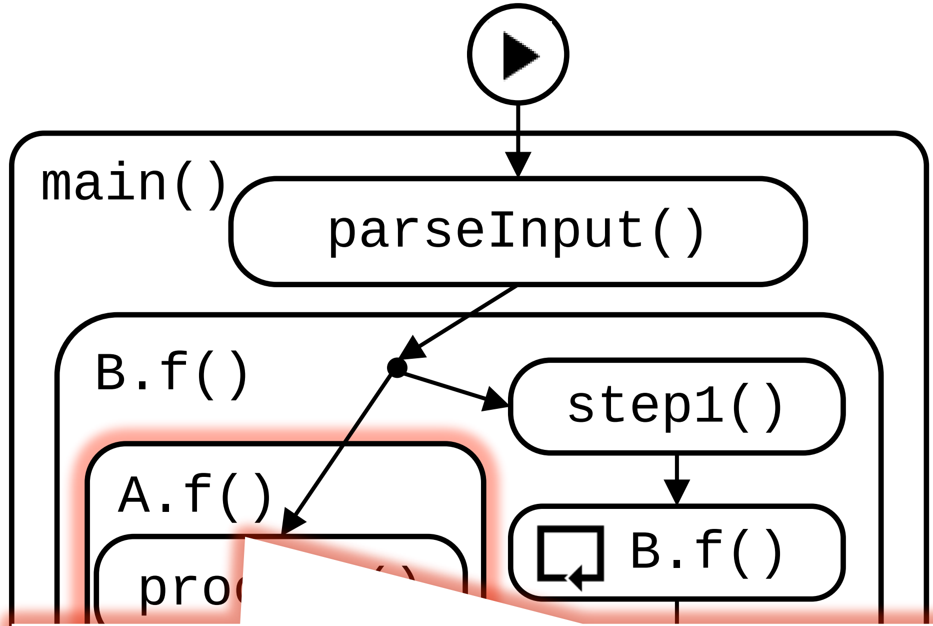


```
main(int i) {
    A a = parseInput();
    a.f(i);
    outputResult();
}
Class A {
    f(int i) {
        process(i);
    }
}
Class B extends A {
    f(int i) {
        if (i == 0) {
            per.f(i);
        e {
            ep1();
            i-1);
            ep2();
```

Logged Event:

Case	Software Run 1
Activity	org.package.ClassA.main(...) + return
Time	04-10-2016 T 12:00:00.126
Resource	thread-1
Source	ClassA.java @ line 25

Bridge between model and code



```
main(int i) {
    A a = parseInput();
    a.f(i);
    outputResult();
}
Class A {
    f(int i) {
        process(i);
    }
}
Class    extends A {
    f(i    i) {
          (i == 0) {
             per.f(i);
             e {
             ep1();
             i-1);
             ep2();
```

Logged Event:

Case	Software Run 1
Activity	org.package.ClassA.main(...) + turn
Time	04-10-2016 T 12:00:00.126
Resource	thread-1
Source	ClassA.java @ line 25

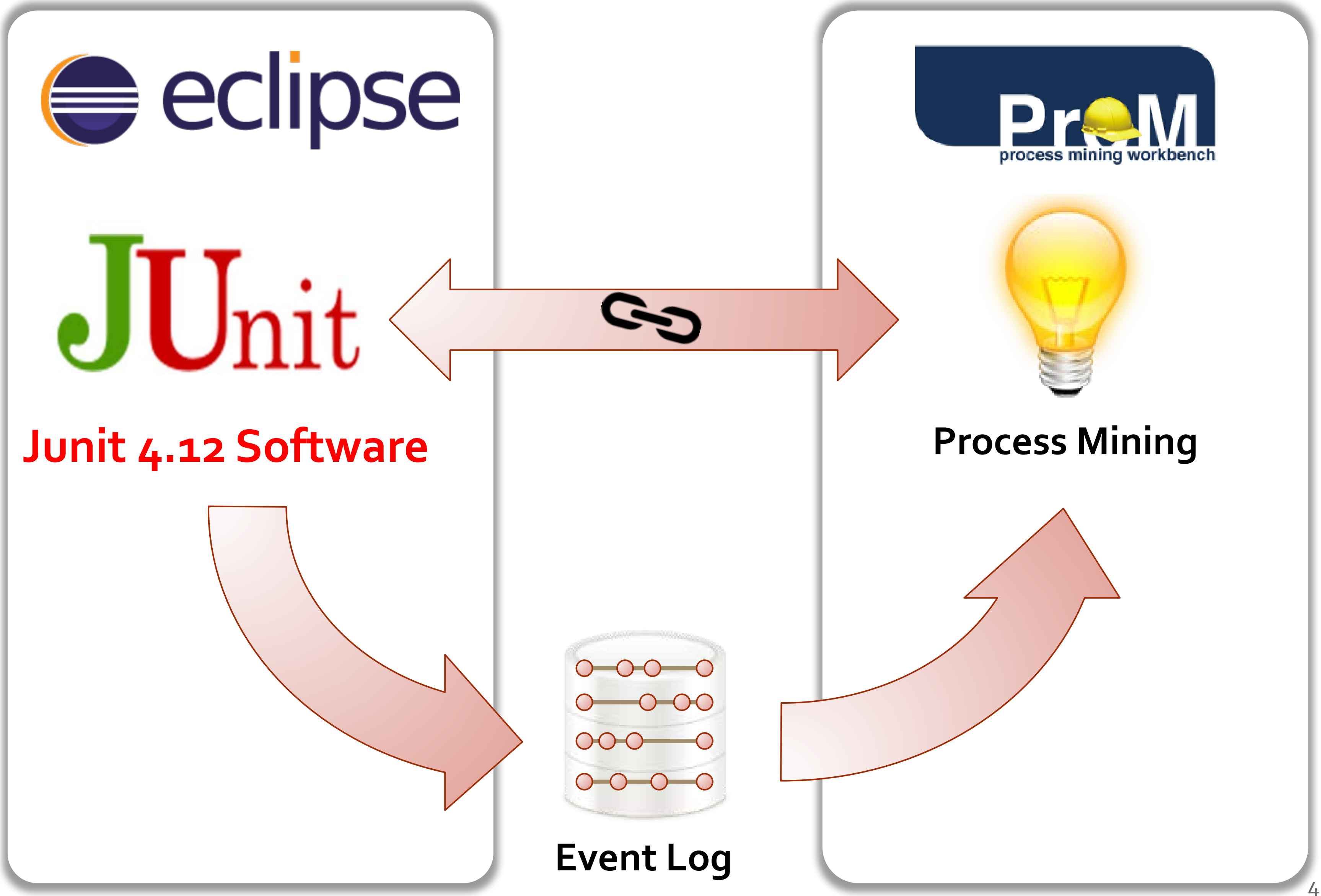
Mining Software in Practice

Junit 4.12 Software

Question 1:
What is the behavior?

Question 2:
How is it used?

Mining Software in Practice



Junit 4.12 Software

Process Mining

Event Log

Challenges

Rich behavior discovery support

- Control flow and synchronization in concurrent systems
- Exceptional and error-driven control flow

Challenges

Rich behavior discovery support

- Control flow and synchronization in concurrent systems
- Exceptional and error-driven control flow

Extend to streaming & monitoring context

- In vivo (online) analysis

Challenges

Rich behavior discovery support

- Control flow and synchronization in concurrent systems
- Exceptional and error-driven control flow

Extend to streaming & monitoring context

- In vivo (online) analysis

Software health and performance

- Reliability analysis based on behavior and performance

Challenges

Rich behavior discovery support

- Control flow and synchronization in concurrent systems
- Exceptional and error-driven control flow

Extend to streaming & monitoring context

- In vivo (online) analysis

Software health and performance

- Reliability analysis based on behavior and performance

Utilize documentation, models and source code

- Capture domain knowledge; even (partial) model descriptions

Try it out yourself

https://svn.win.tue.nl/repos/prom/XPort/
https://svn.win.tue.nl/repos/prom/Packages/Statechart/

Maikel Leemans m.leemans@tue.nl