

OPEN SOURCE
SERVER WHICH
EVERY DEVELOPER
WANTS TO USE

X



JENKINS^X

OPEN SOURCE AUTOMATION SERVER





ธนา ตั้งประสม



สุมินชา ชลอวงษ์



ปรัชญา ทองแสน



ปรัตถกร ศรีบรรยงค์



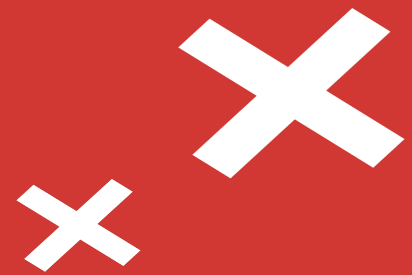
พิม ปิยจิรานันท์



ภากรณ์ ธนประชานันท์

สมาชิกกลุ่ม

JENKINS



What is Jenkins?

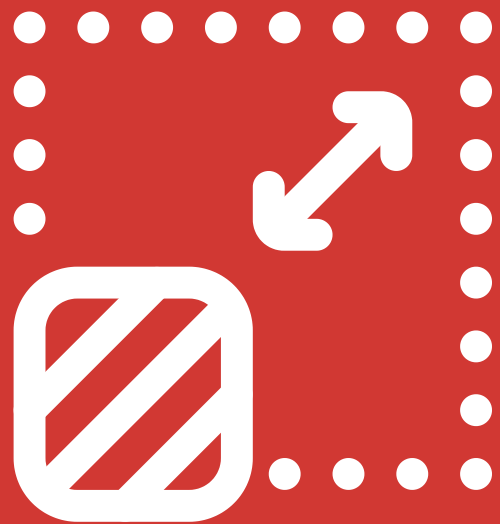
Jenkins is an easy-to-use open-source java-based CI/CD tool. It has been around for some time, and several organizations use it for their CI/CD needs.

Jenki

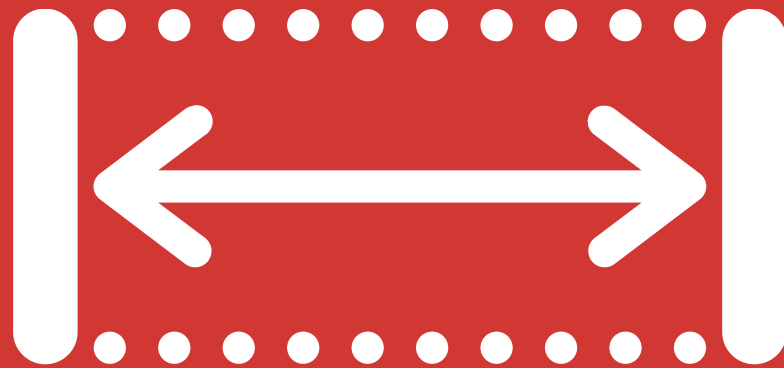
INS

Quality Attribute (QA)

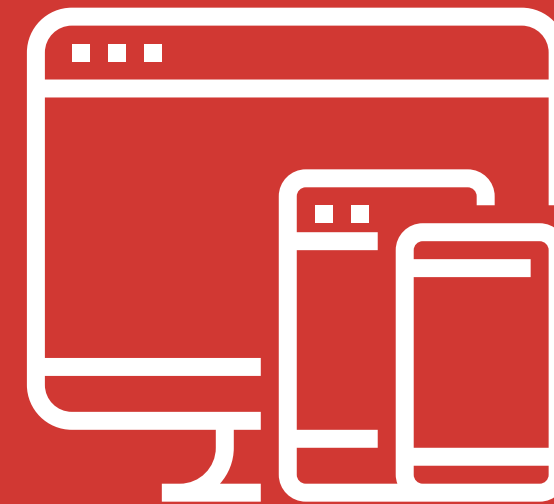
"Quality attribute is a measurable or testable property of a system that is used to indicate how well the system satisfies the needs of its stakeholders." Quality attribute in Jenkins consist of...



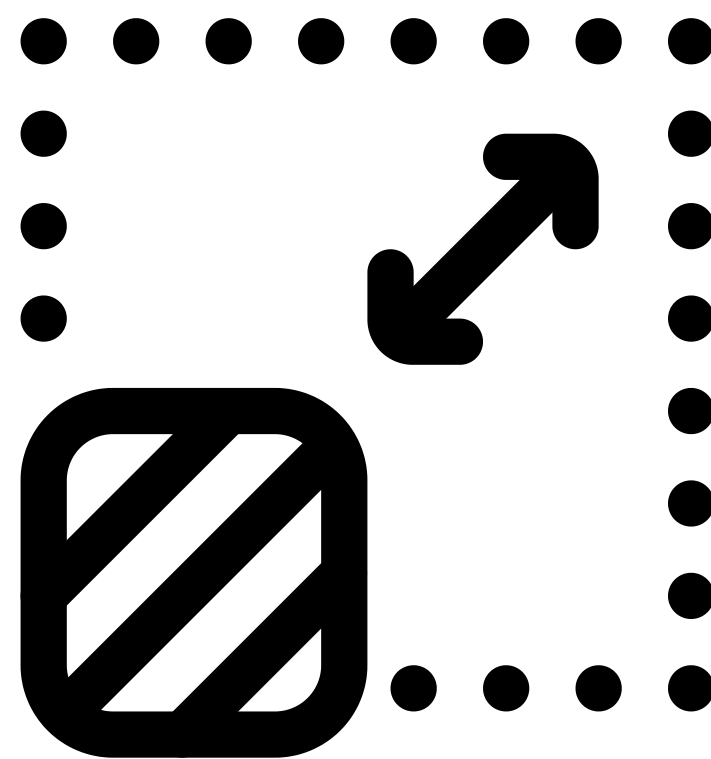
Scalability



Extensibility

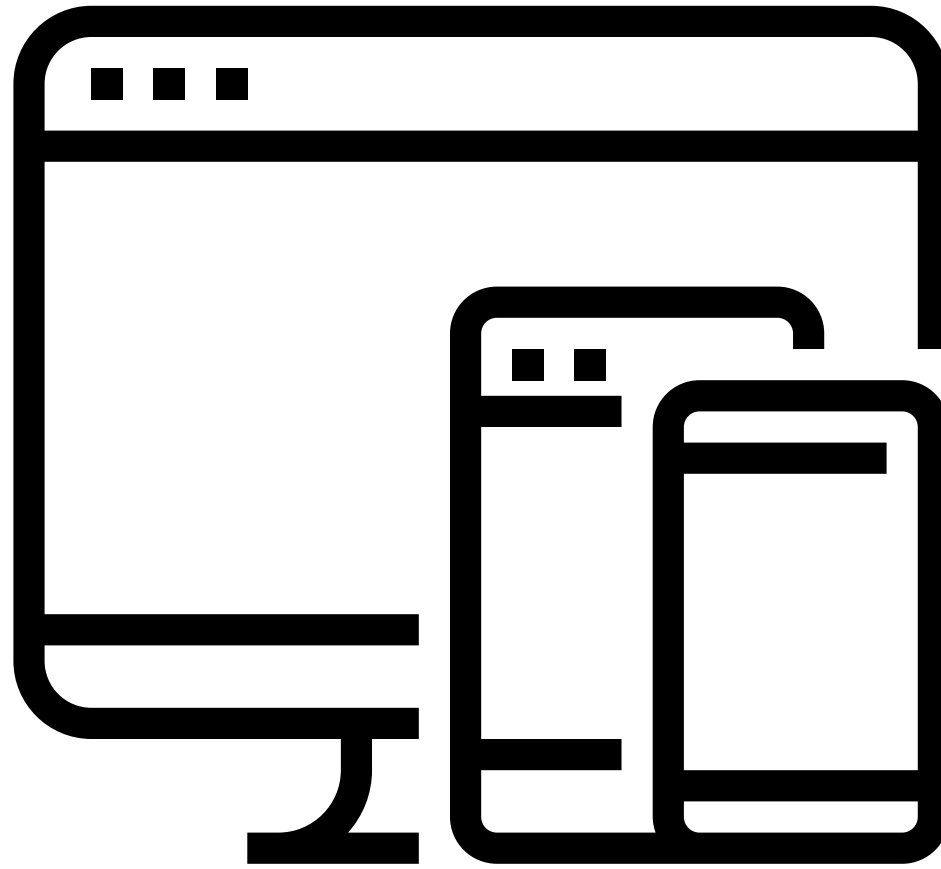


Portability



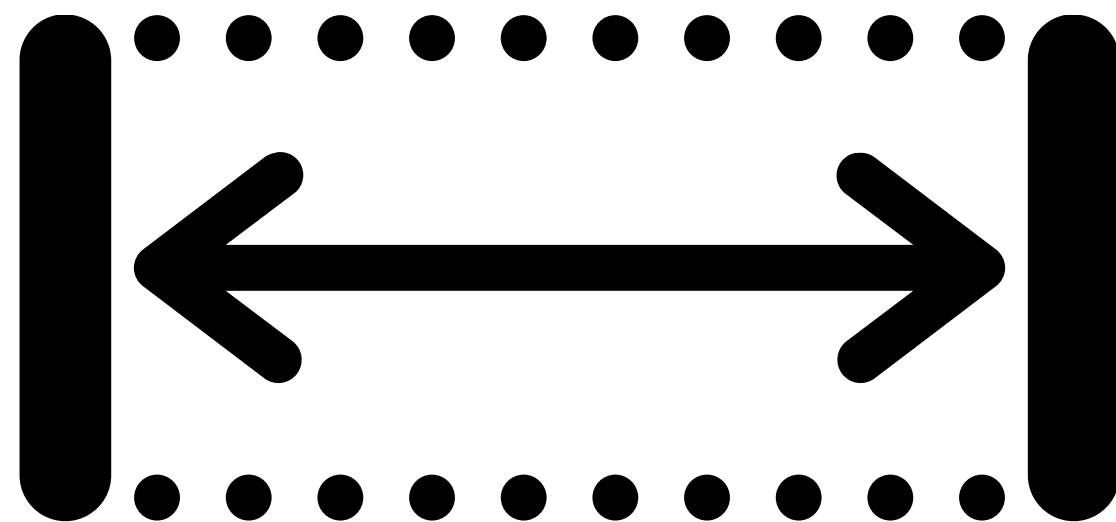
Scalability

นักพัฒนาเลือกใช้ QA นี้เพราะว่า Jenkins นั้น Support Master-Slave และถูกออกแบบให้มีการใช้งาน protocol TCP/IP ในการติดต่อสื่อสารระหว่างกันเองของ Jenkins เพื่อให้สามารถทำตามคำสั่งเมื่อมีการสั่งงานมาจาก Master-Node



Portability

Jenkins ต้องสามารถใช้งานได้ในทุก Platform อีกทั้ง Jenkins เขียนด้วยภาษา Java ที่รันบน JVM ทำให้สามารถ Cross platform ได้



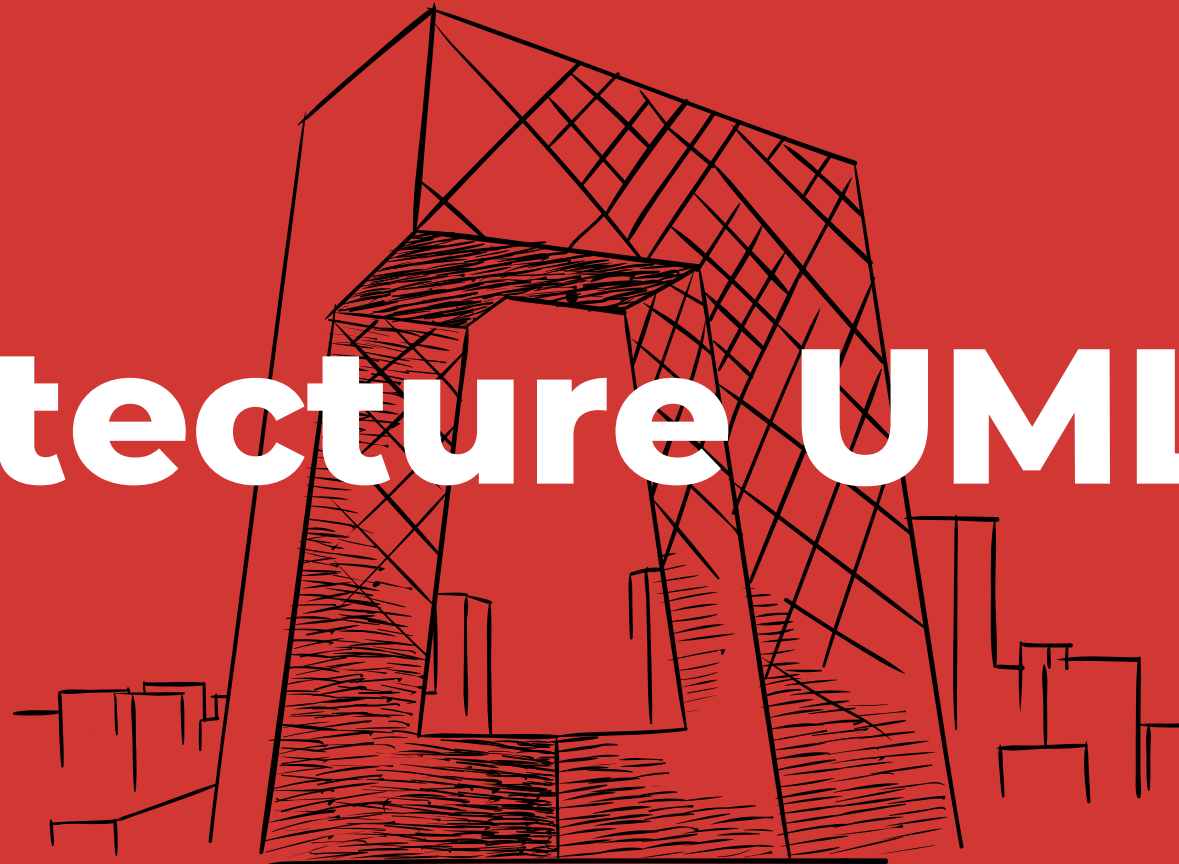
Extensibility

นักพัฒนาเลือกใช้ QA นี้เพราะว่า Jenkins object model สามารถที่จะ extend ได้ (เช่นเราสามารถที่จะเพิ่ม SCM implementation จาก interface ที่ jenkins ให้มา) อีกทั้ง Jenkins จะให้ interface มาเพื่อให้นักพัฒนามาพัฒนาต่อได้

JENKINS



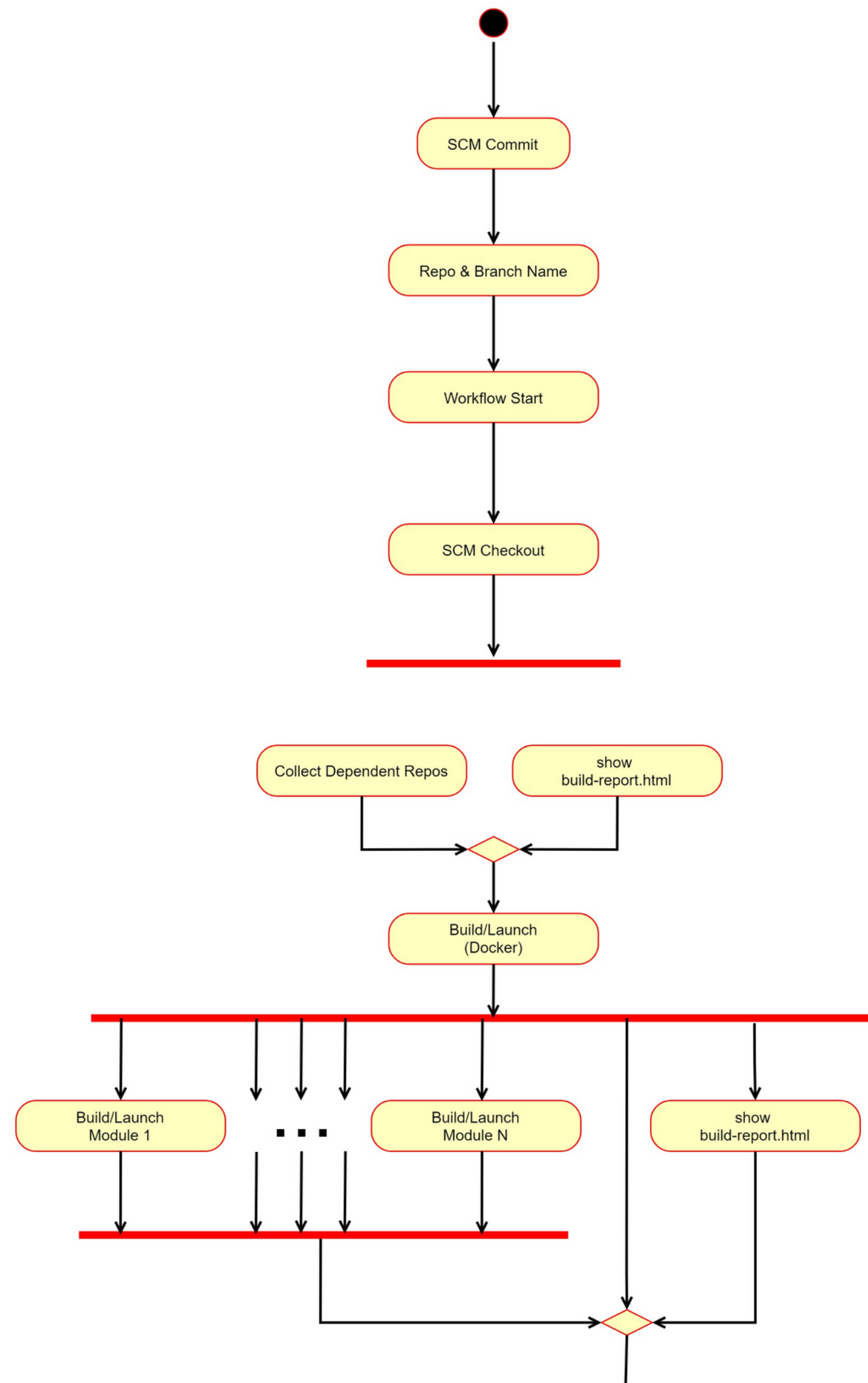
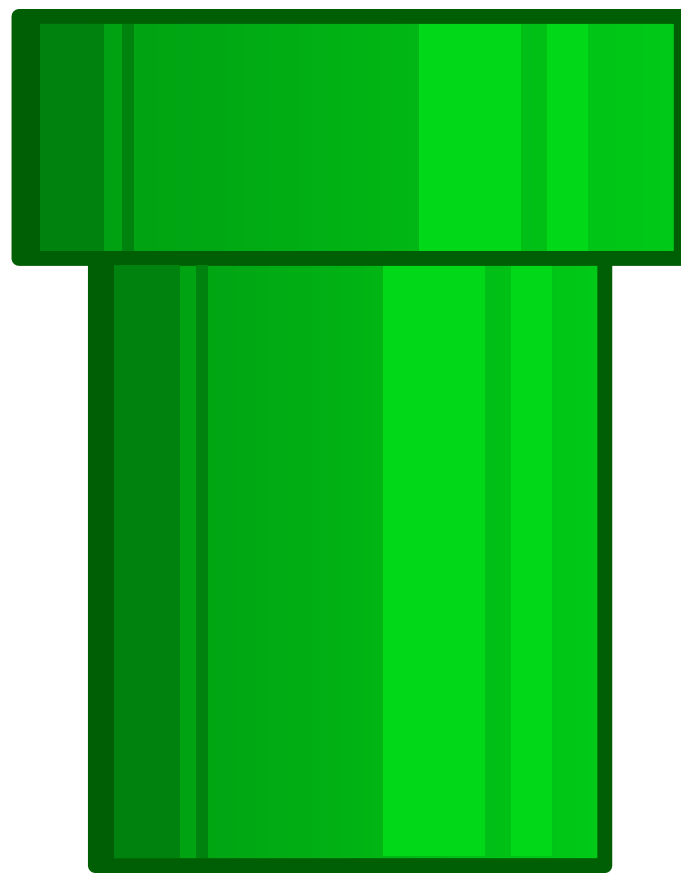
Architecture UML

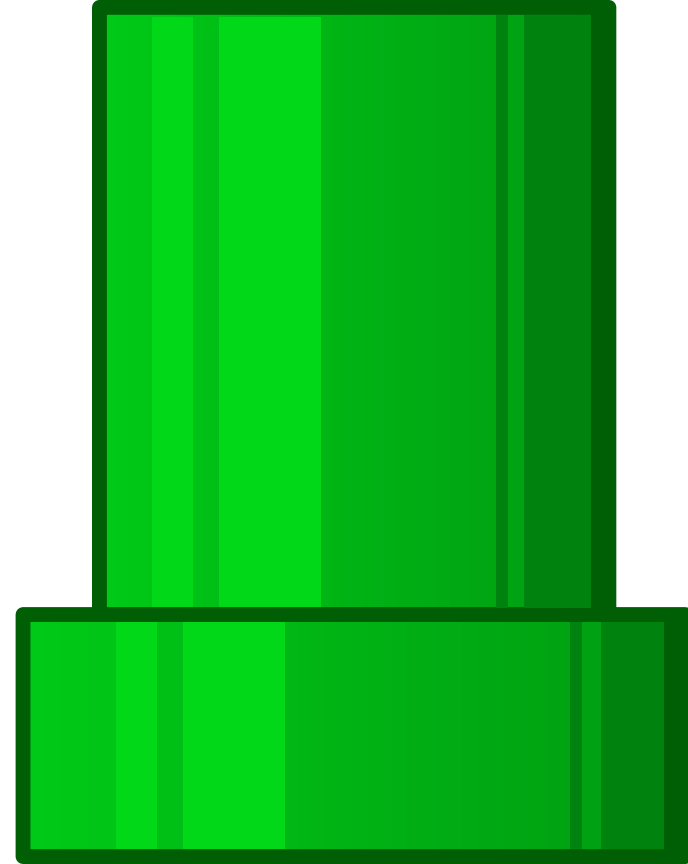


INS

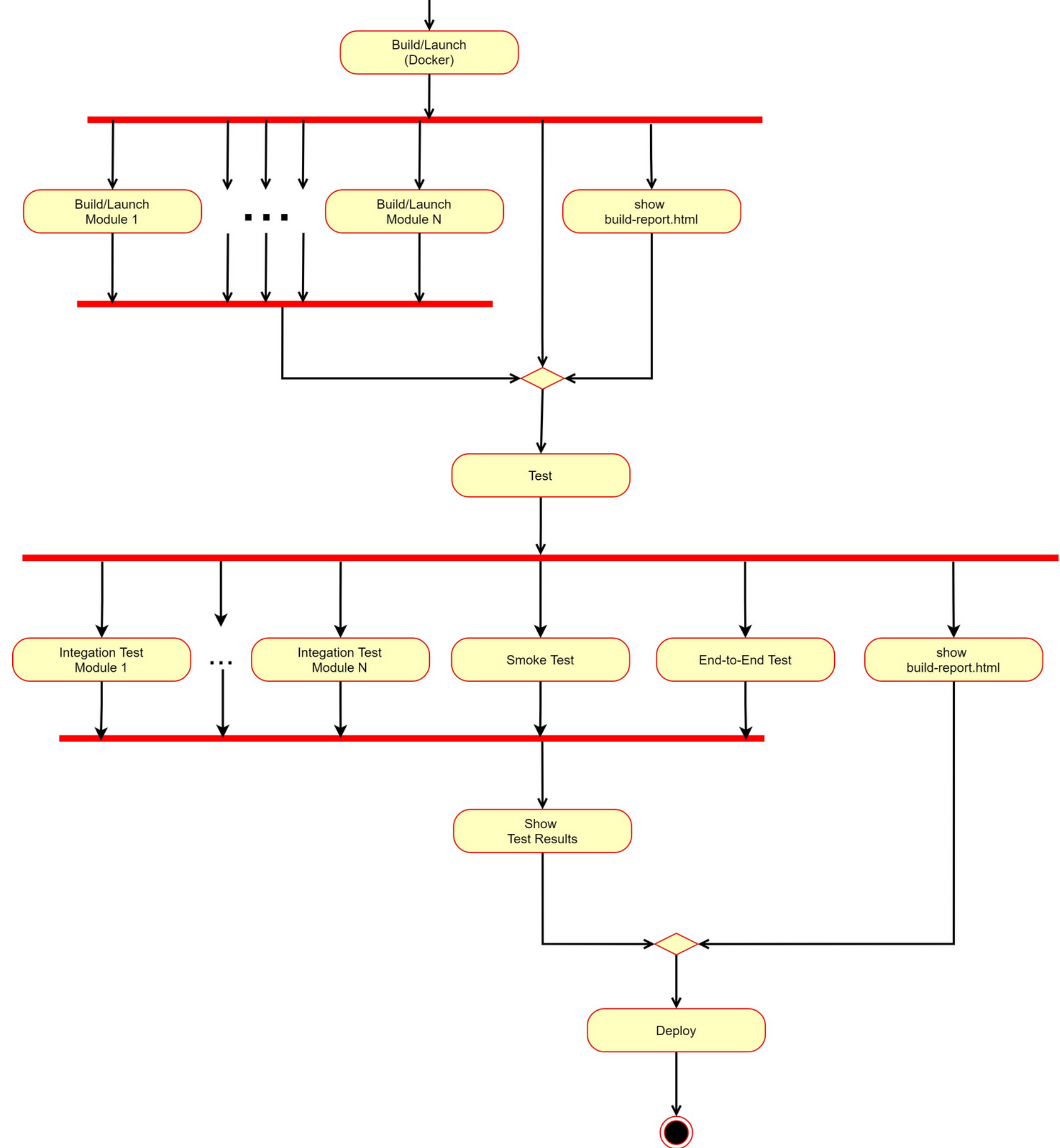


Pipes and Filters





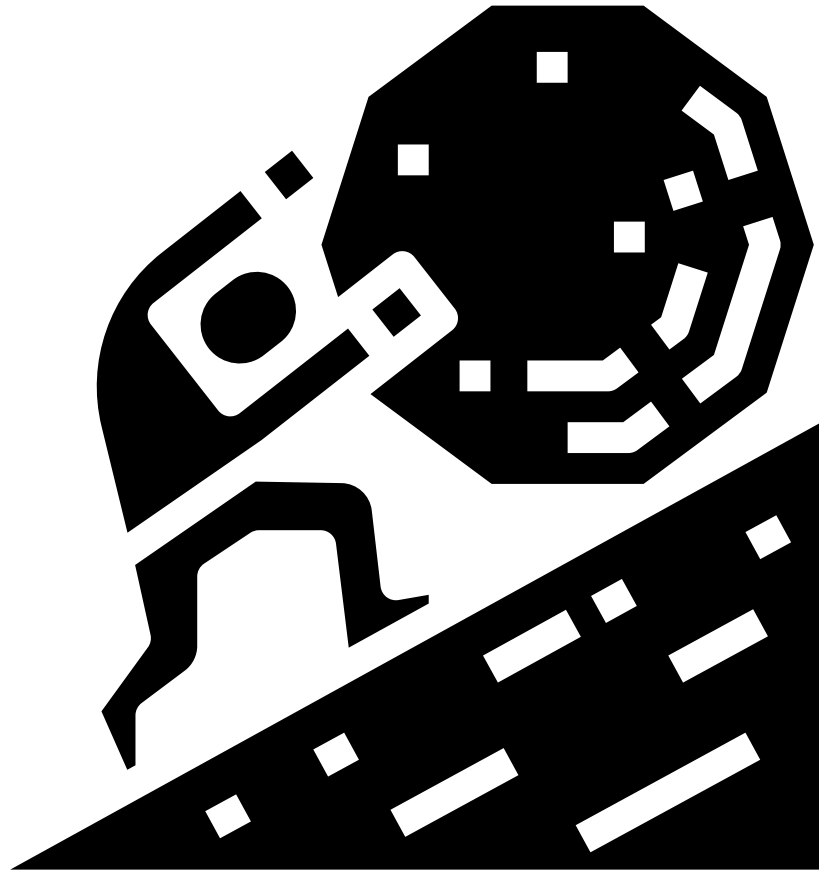
Pipes and Filters (2)





JENKINS

Architecture's weakness



01

Hard to implement Interactive transformations

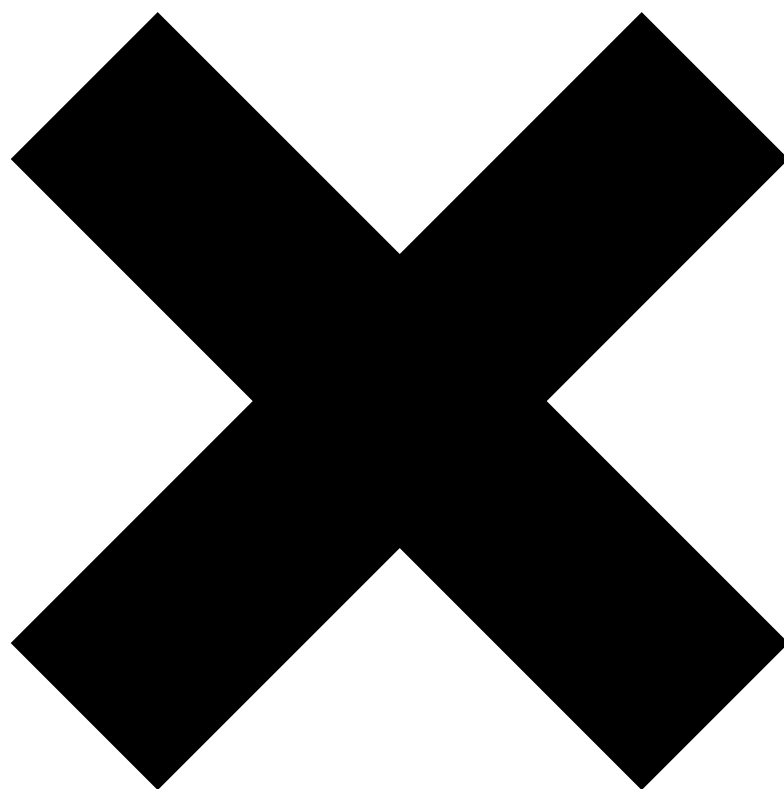
Interactive transformations ทำได้ยาก เนื่องจากตัว architecture เองไม่สนับสนุน interactive system อยู่แล้ว



02

Latency from Computation overhead

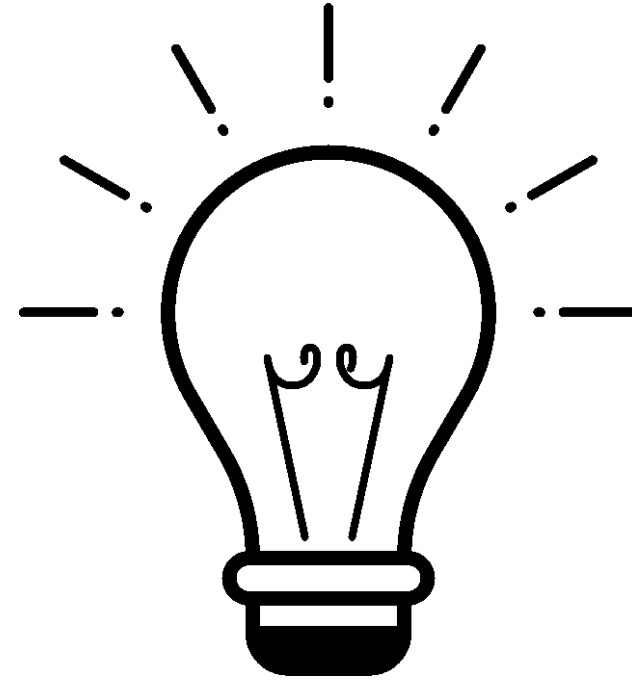
มี Independent filters เยอะทำให้เกิด Computation overhead เกิด Latency



03

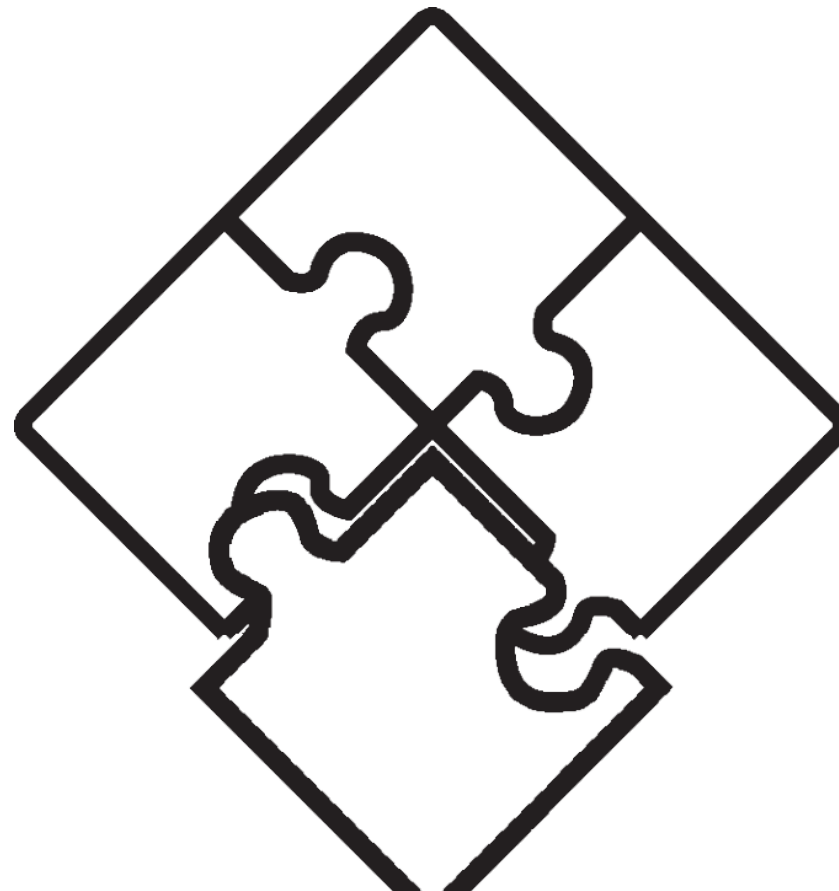
**Not support long-running
computations**

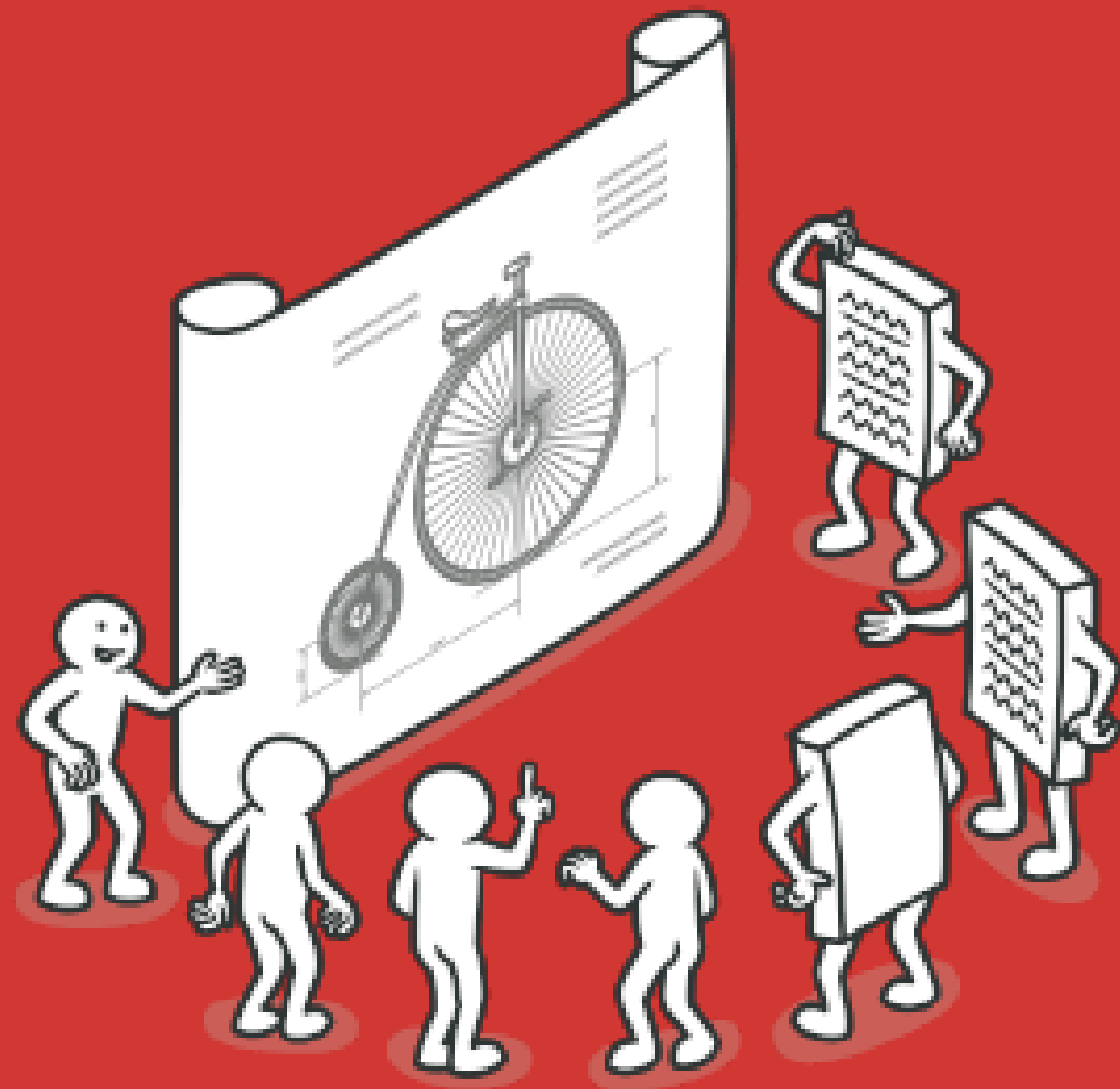
ไม่สนับสนุน long-running computations



Proposed Solution: Increase Throughput

เพิ่ม Throughput ของ Pipes and filter โดยใช้ Task farm parallelization pattern (เป็นการสร้างและจัดการ instances ของ filter แบบ parallel)





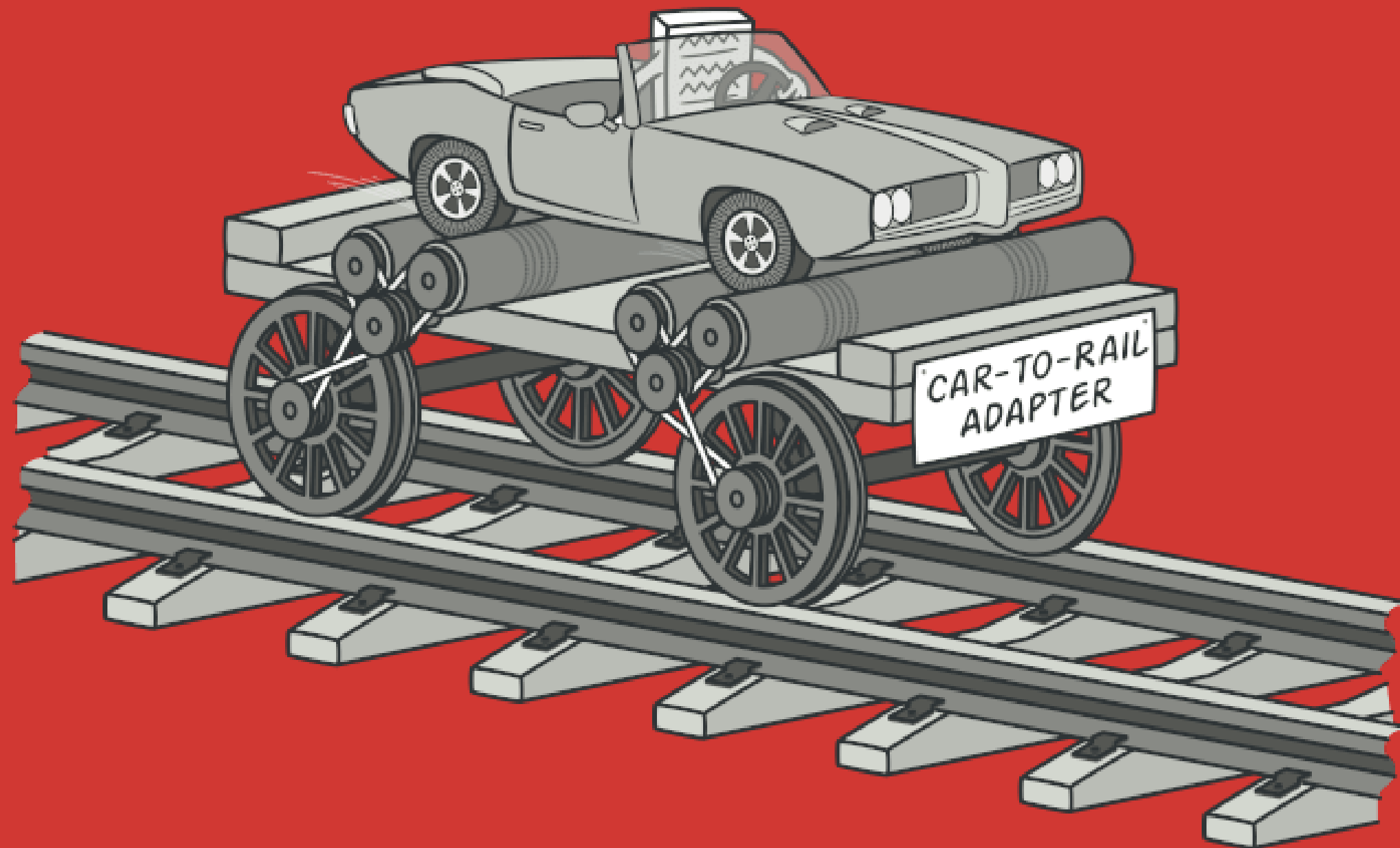
Design pattern

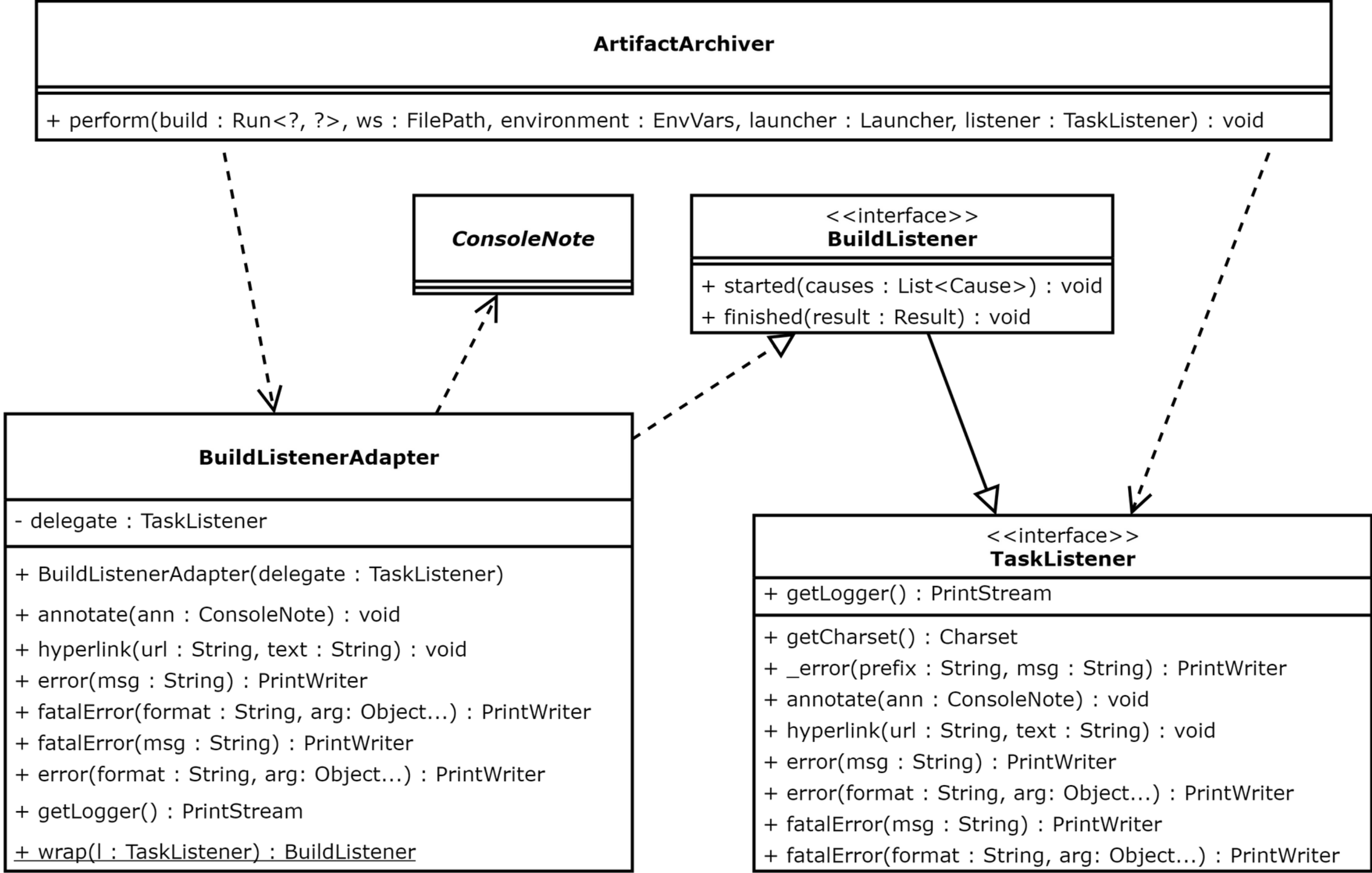
- 01 **Adapter**
- 02 **Singleton**
- 03 **Factory Method**
- 04 **Observer**

01

Adapter

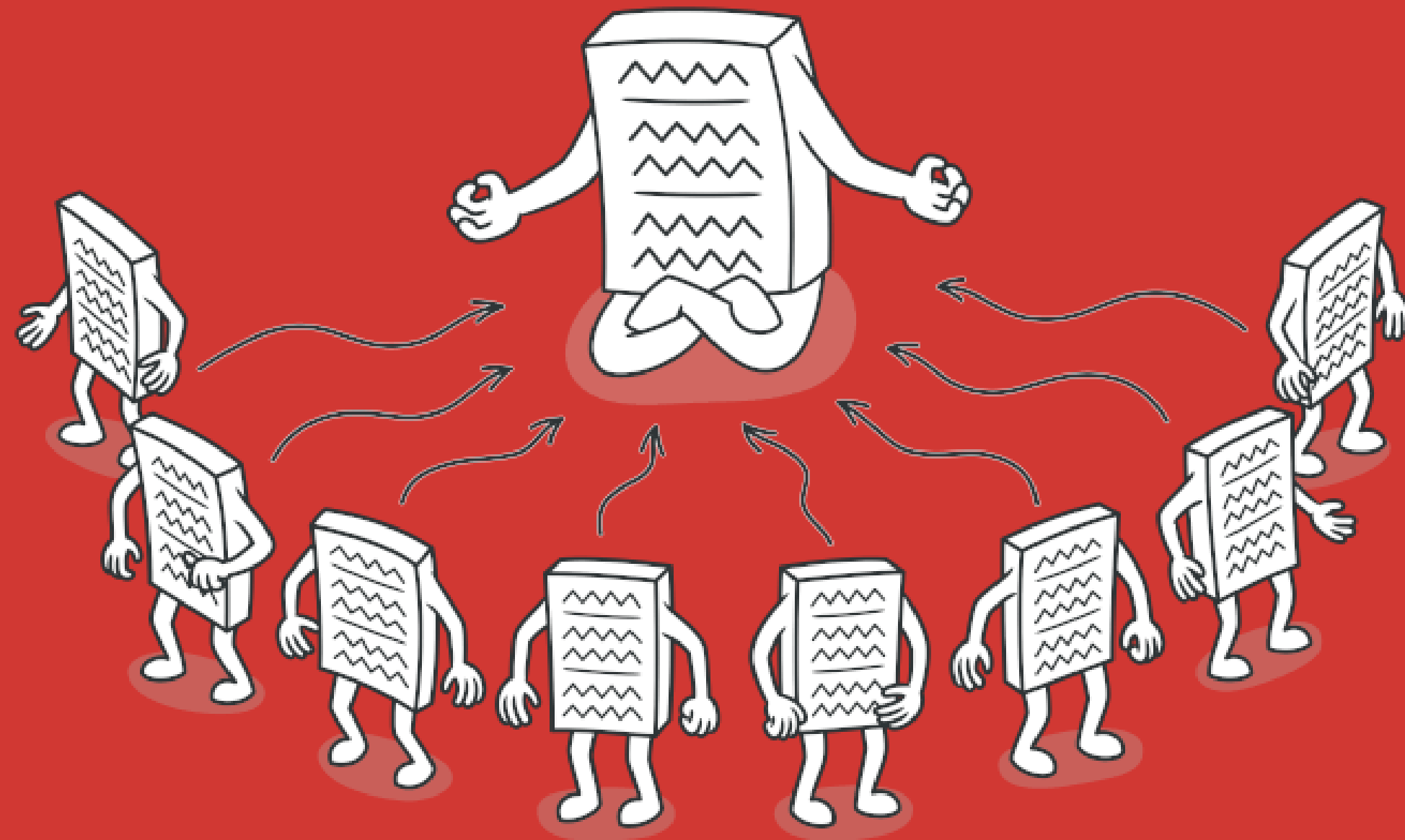
"Adapter is a structural design pattern that allows objects with incompatible interfaces to collaborate."

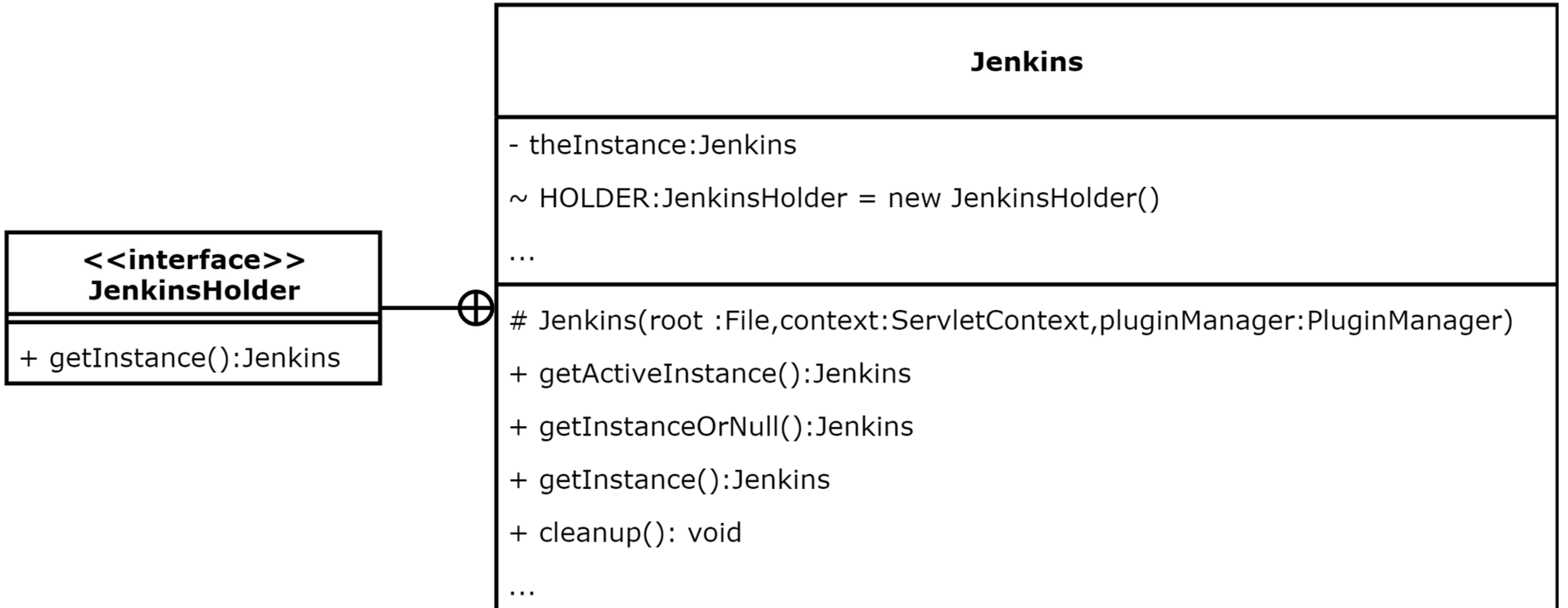




02 Singleton

"Singleton is a creational design pattern that lets you ensure that a class has only one instance, while providing a global access point to this instance."

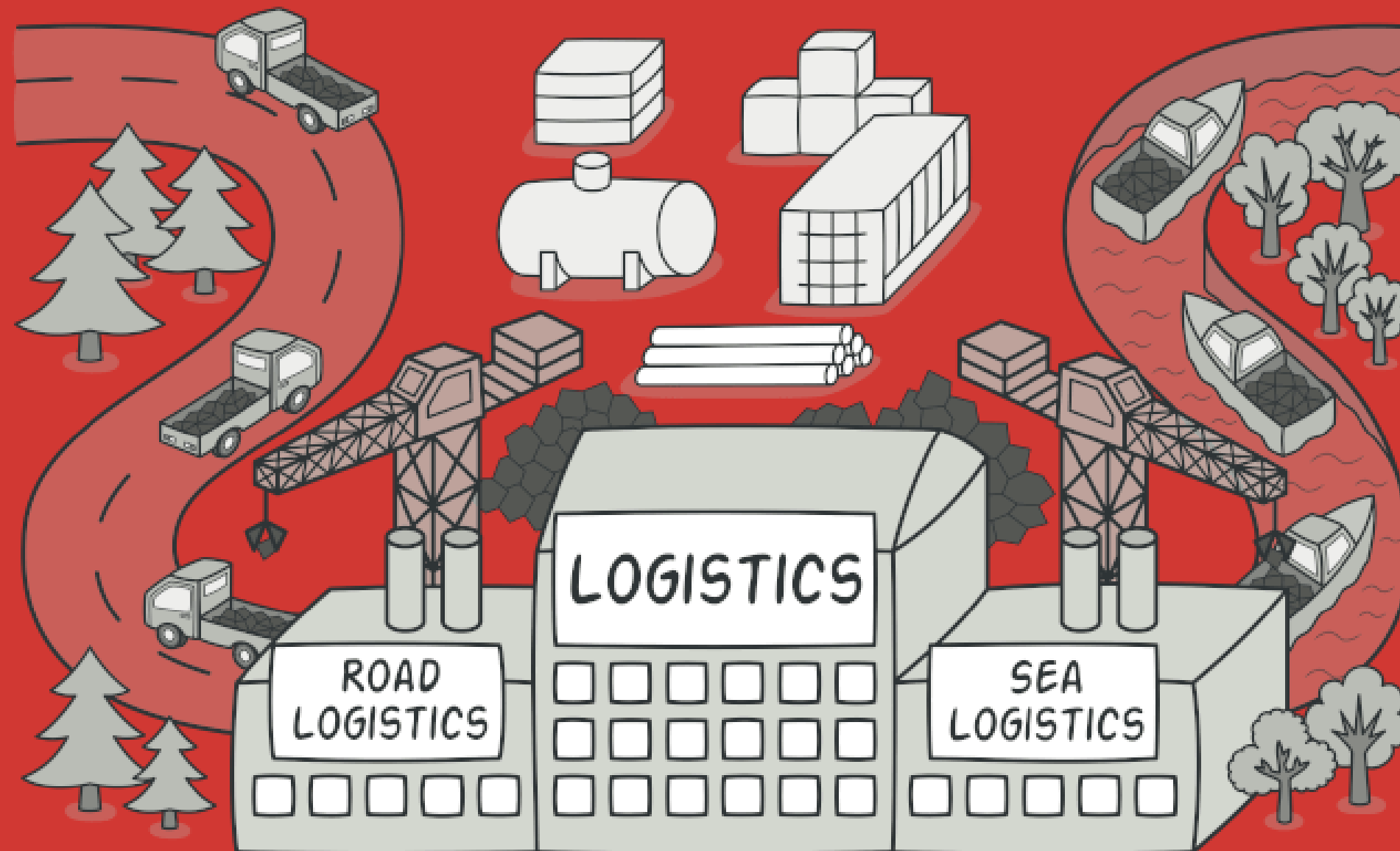


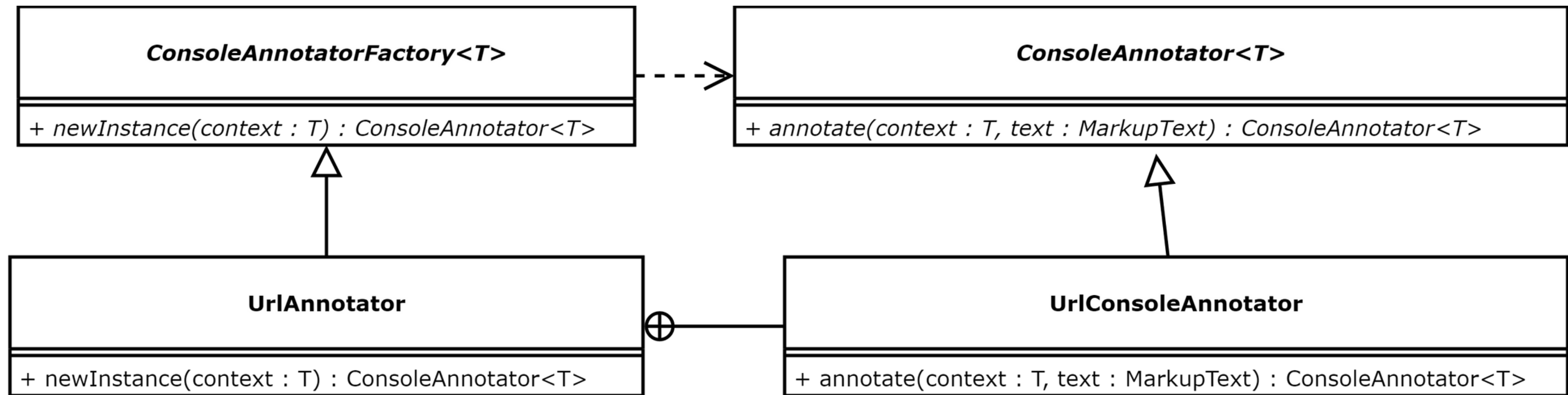


03

Factory Method

"Factory Method is a creational design pattern that provides an interface for creating objects in a superclass, but allows subclasses to alter the type of objects that will be created."

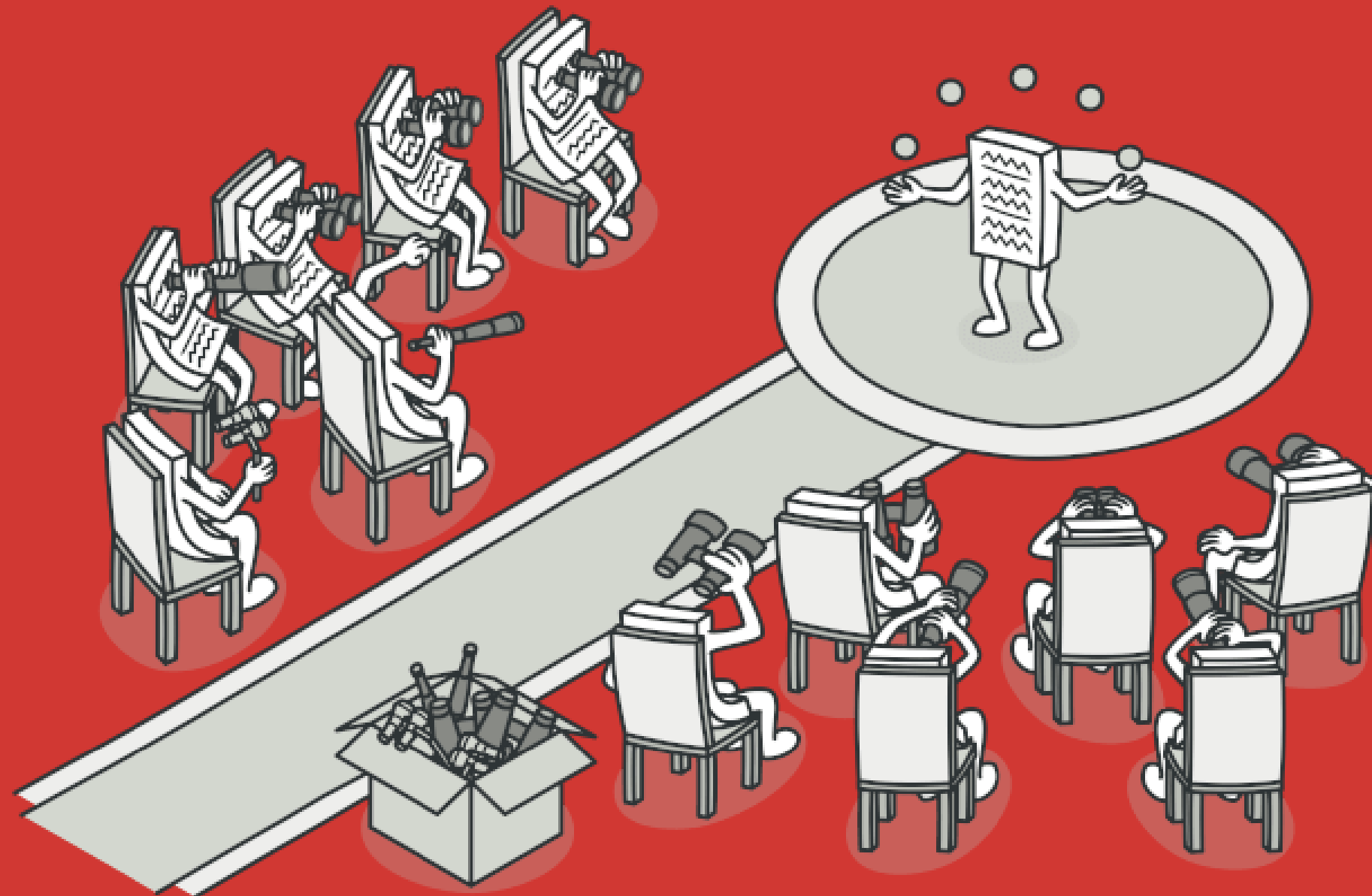


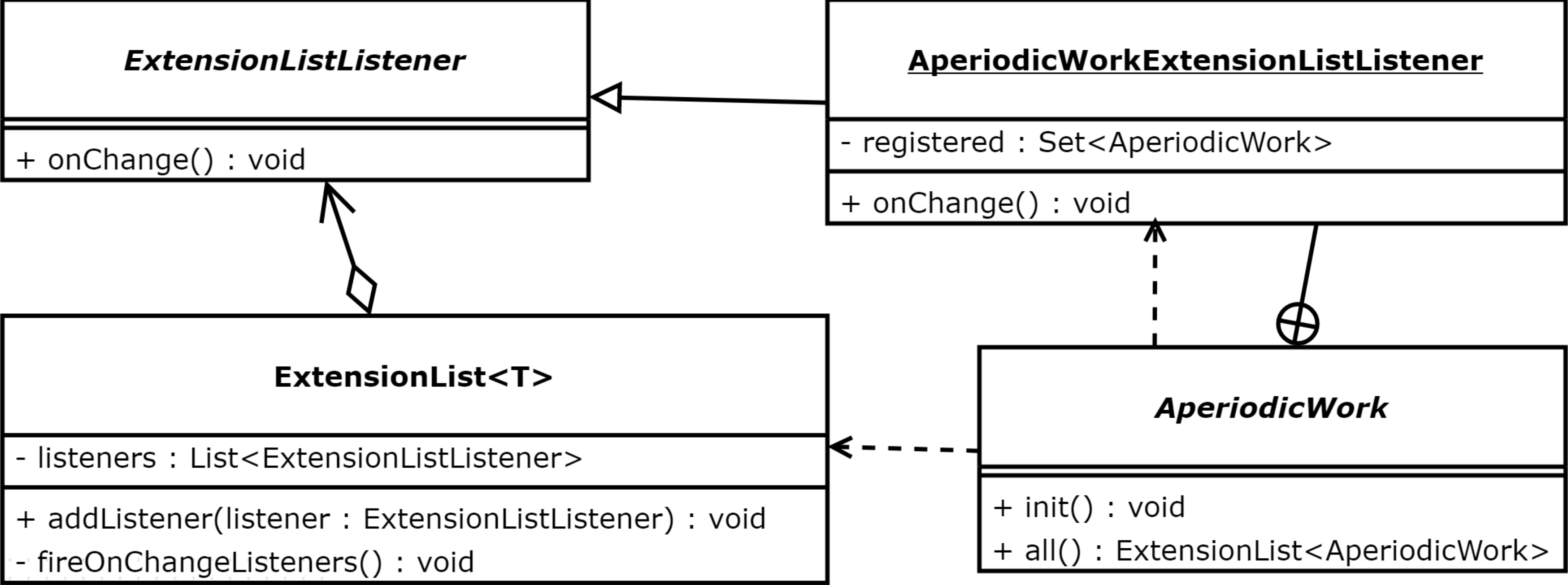


04

Observer

"Observer is a behavioral design pattern that lets you define a subscription mechanism to notify multiple objects about any events that happen to the object they're observing."







**Do you have
any questions
sir?**

