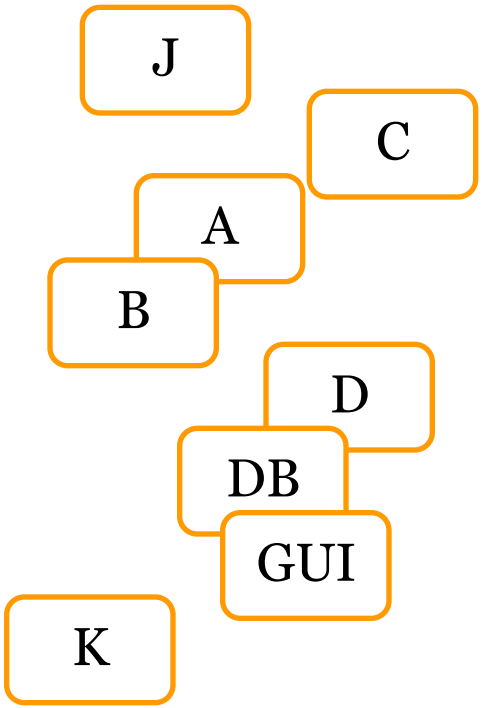
PROMPT Software Testing

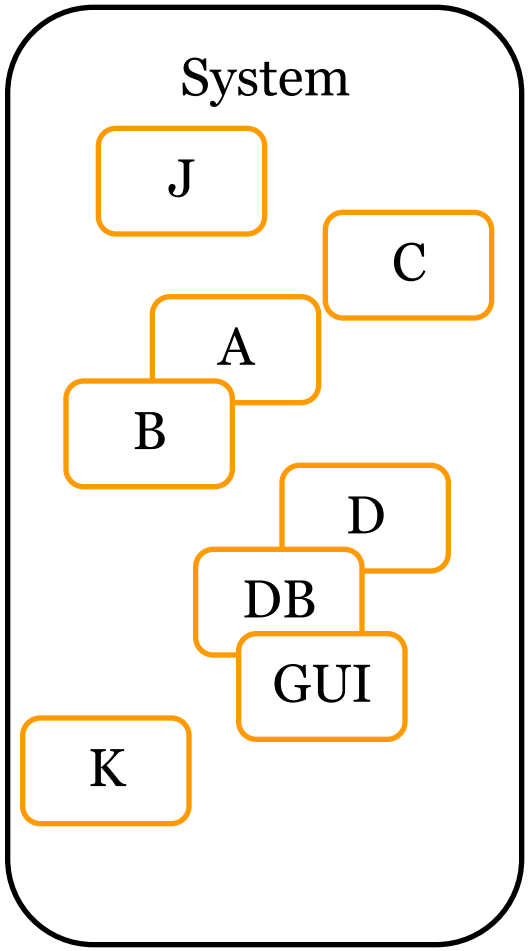
Integration Strategies

Based on the book:
"**Software Testing and analysis**"
by Mauro Pezzé & Michal Young

Integration and integration test

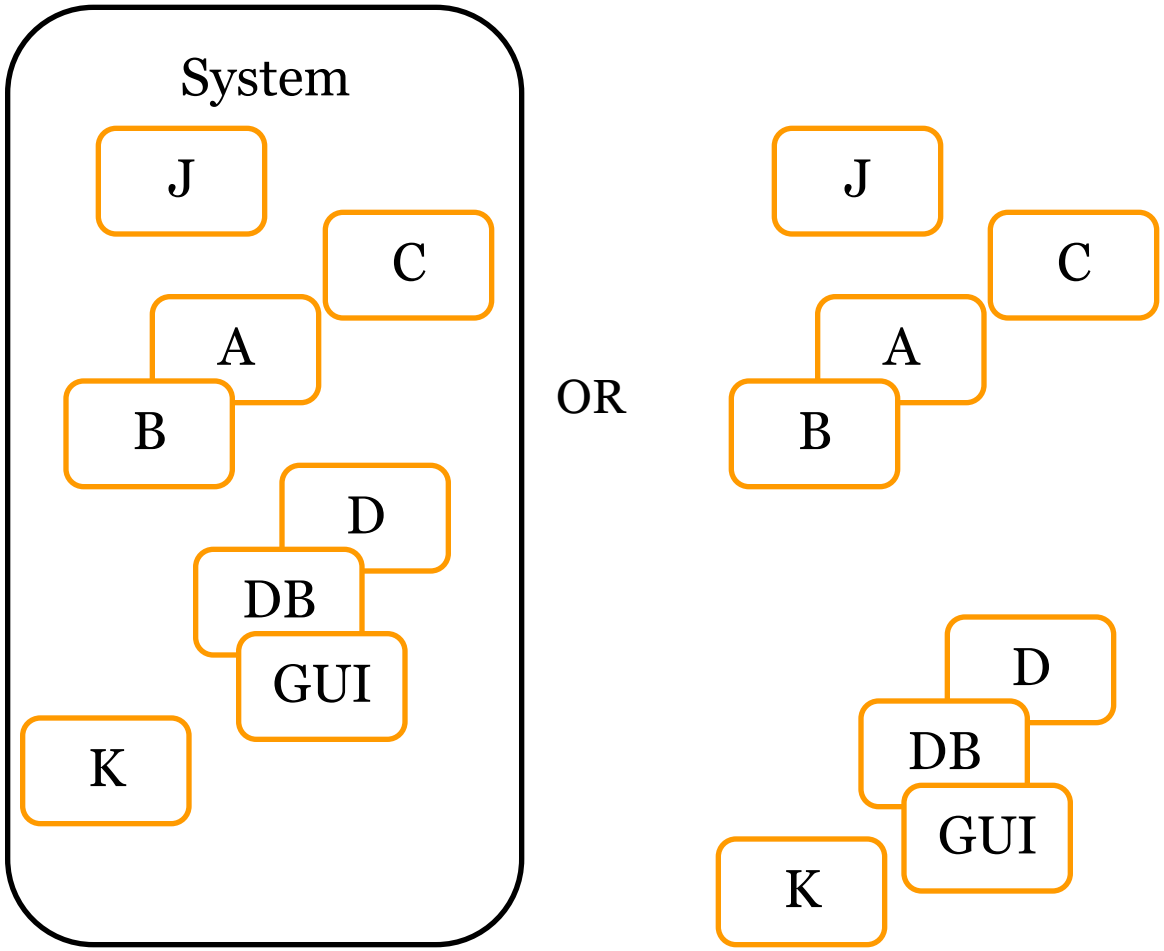


Integration and integration test

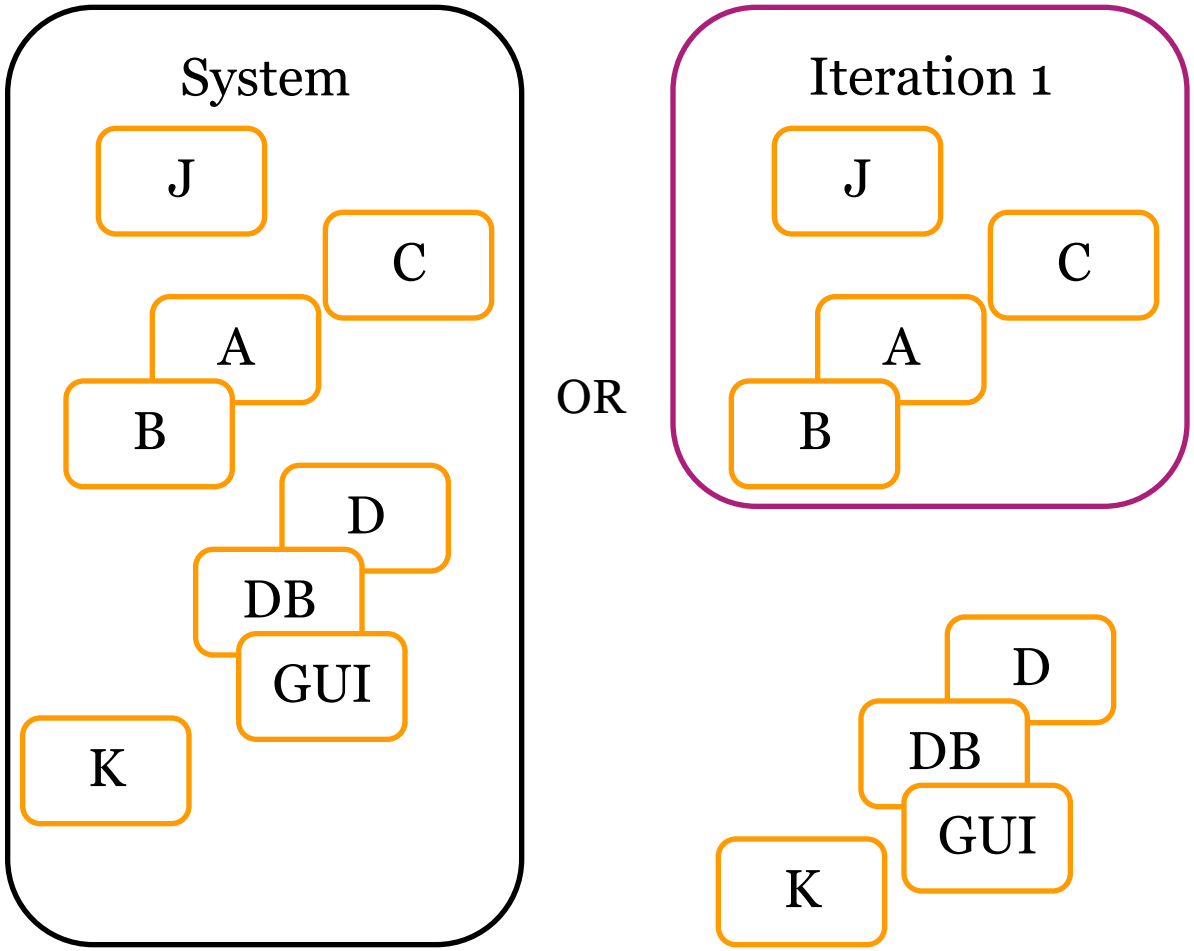


OR

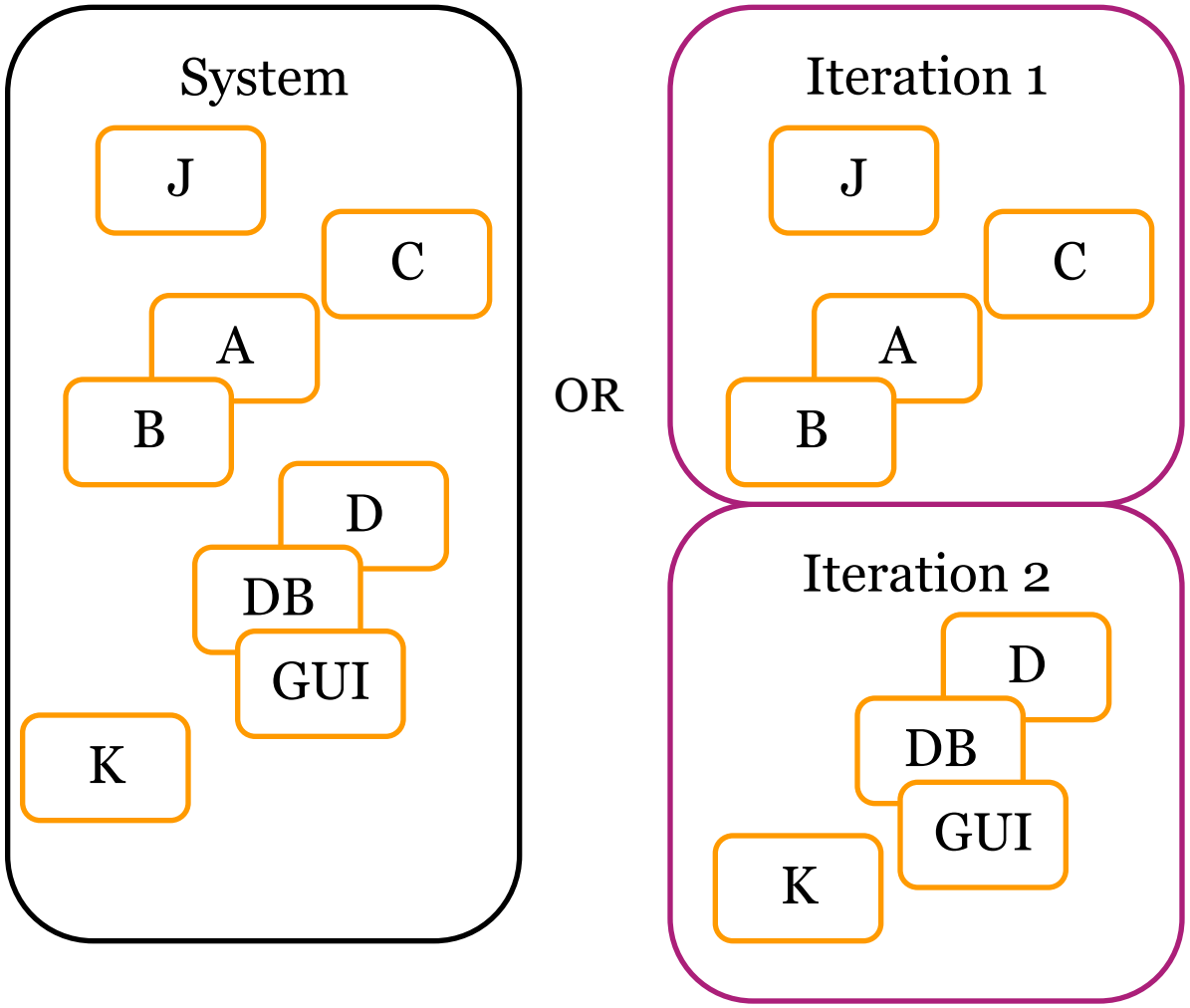
Integration and integration test



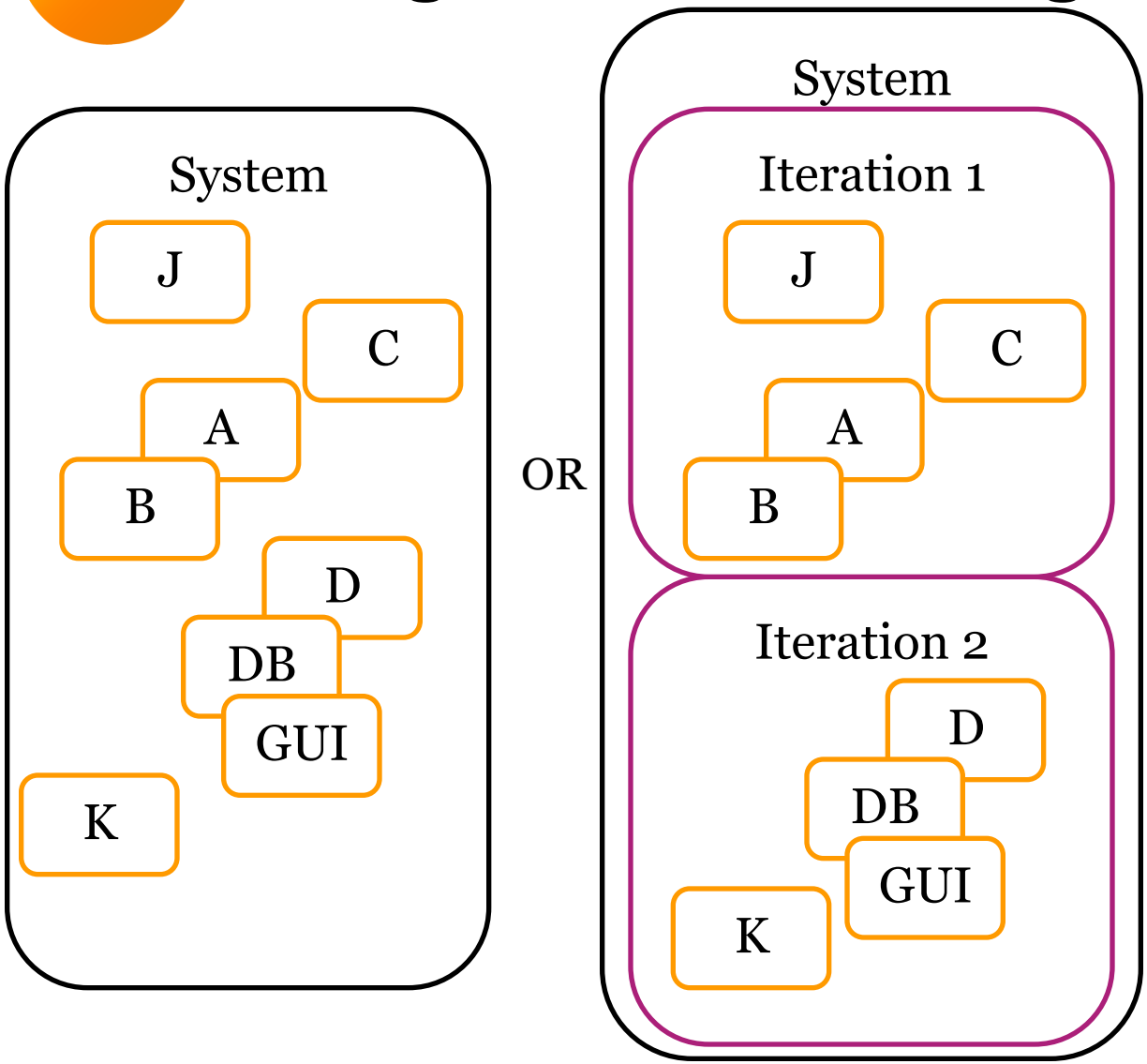
Integration and integration test



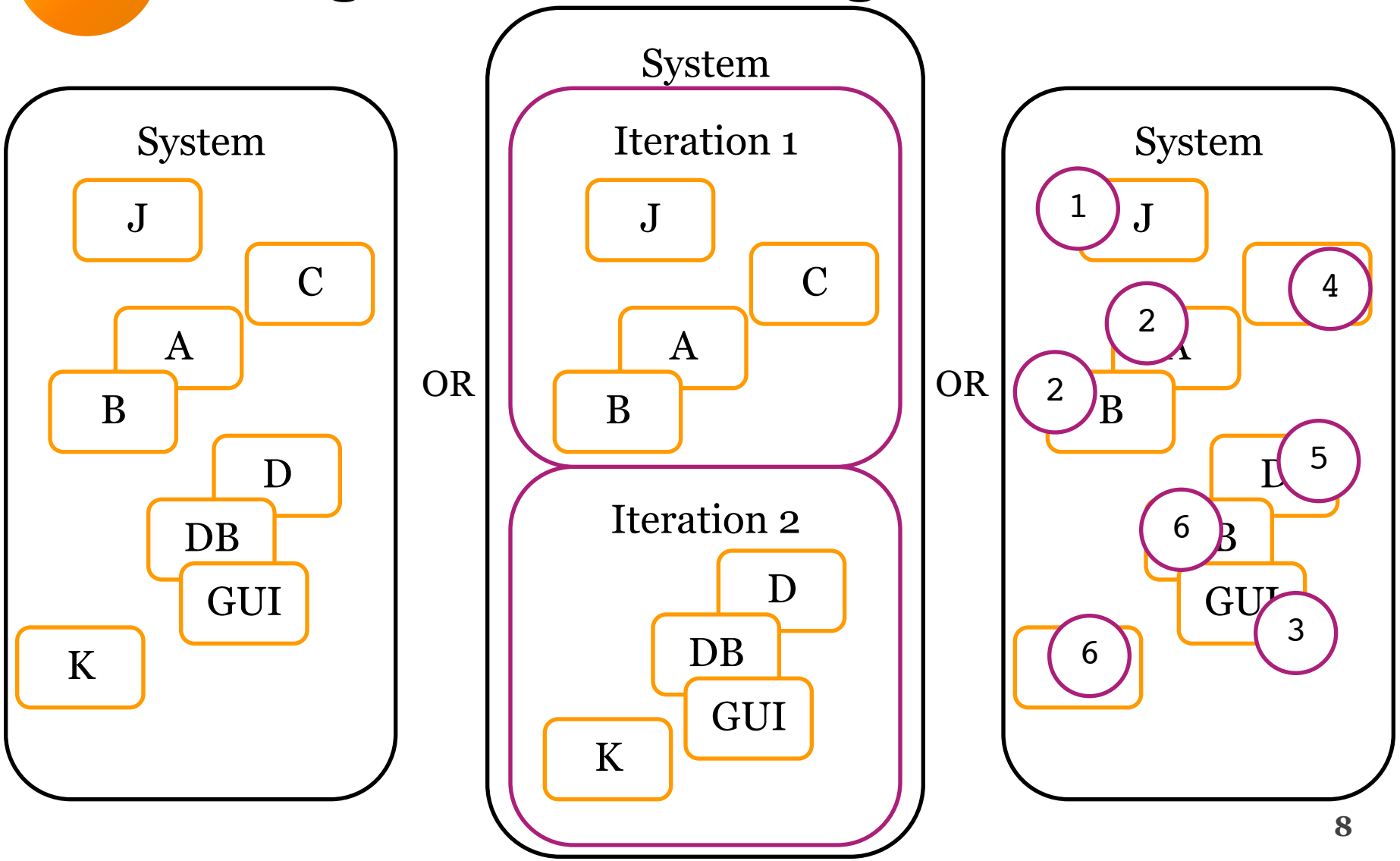
Integration and integration test



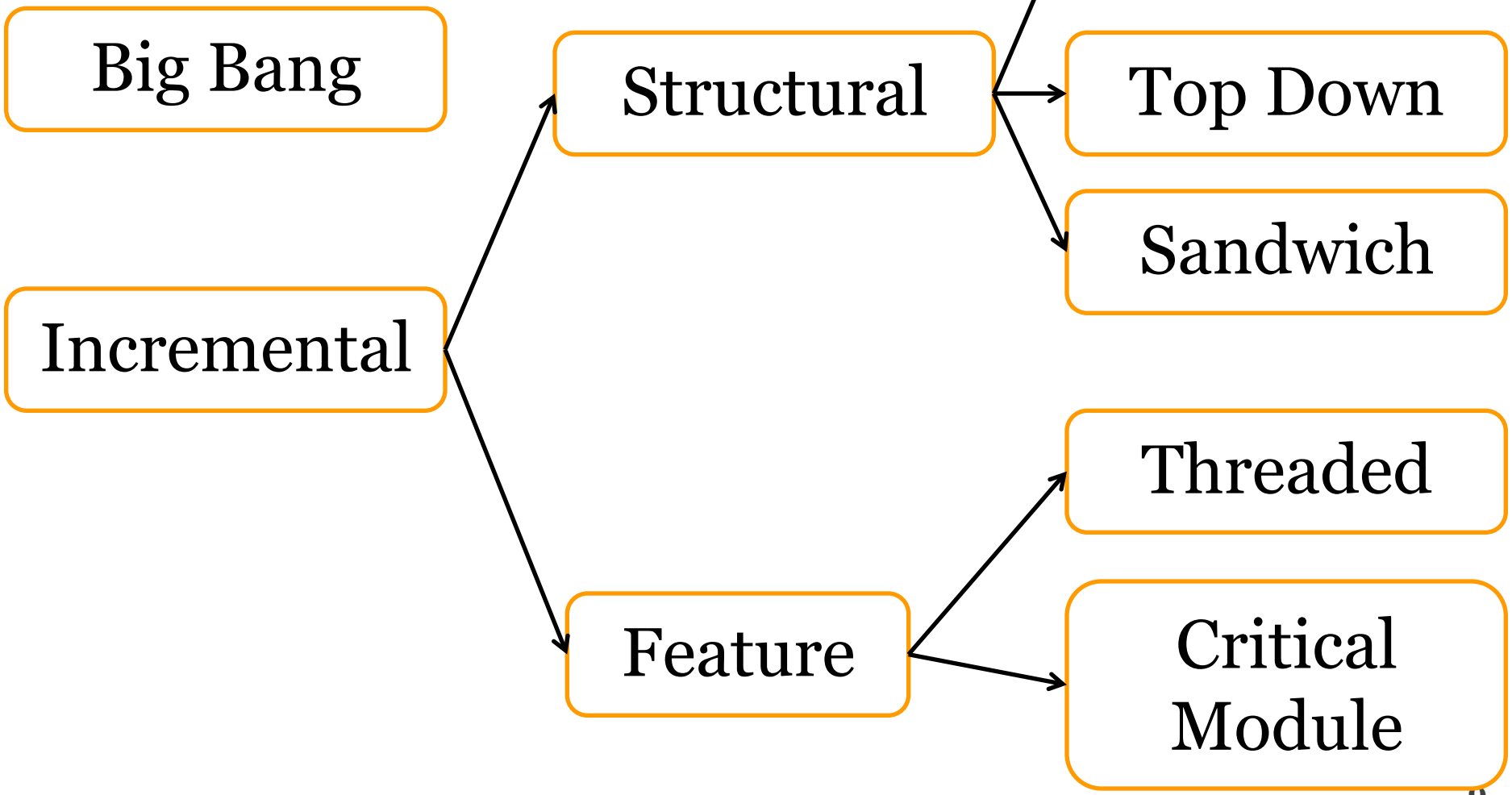
Integration and integration test



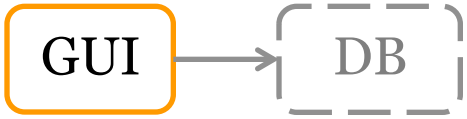
Integration and integration test



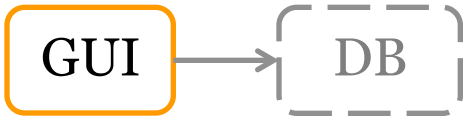
Sample strategies



Faking non existent parts “SCAFFOLDING”



Faking non existent parts "SCAFFOLDING"



Scaffolding (simplified)

Dummy - Only place holders in parameter lists

Fake - Typically quick and dirty implementation

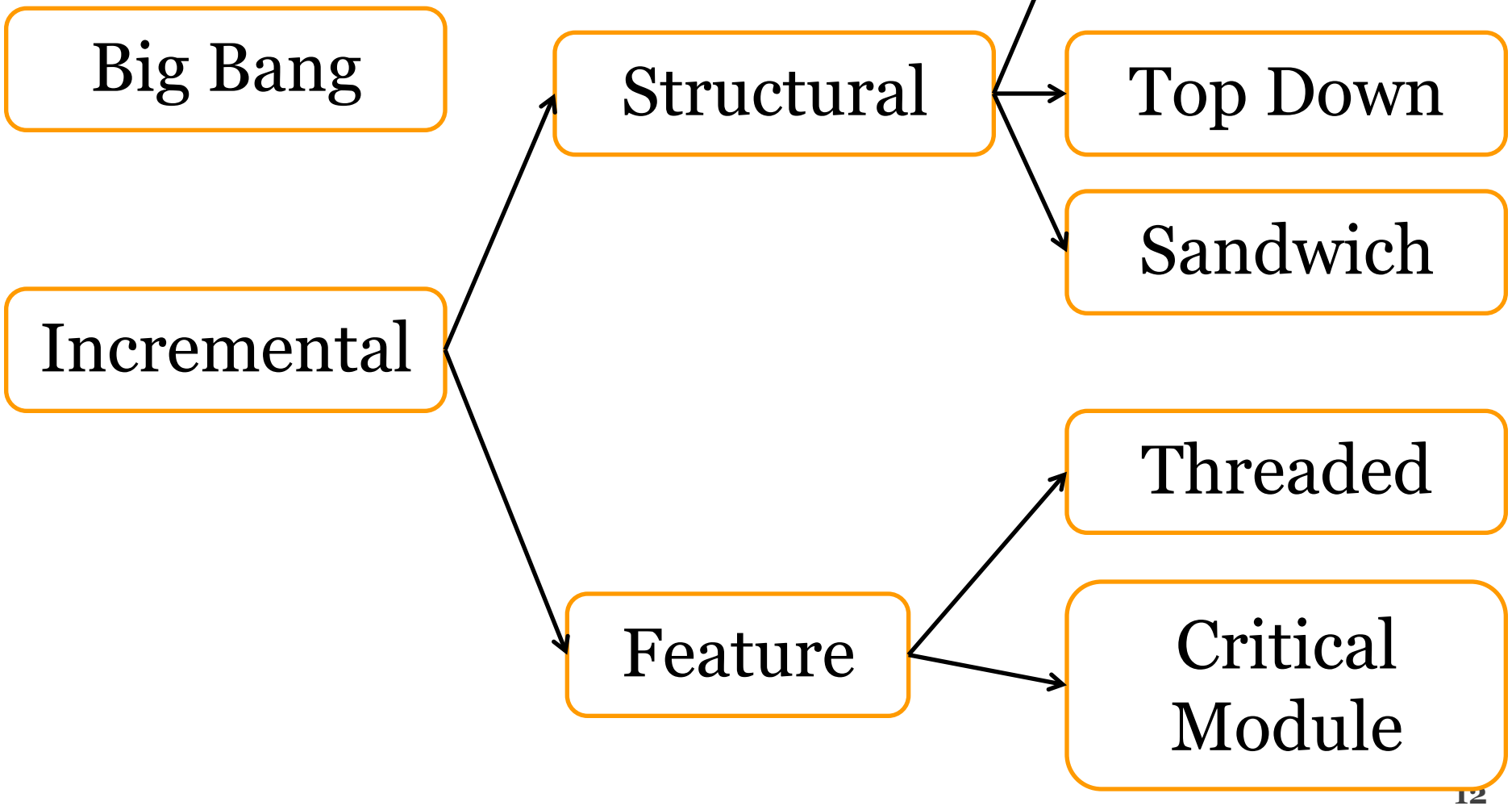
Proxy - Acts if it was "someone" else

Stub - Fixed replies on function calls without any implementation behind.

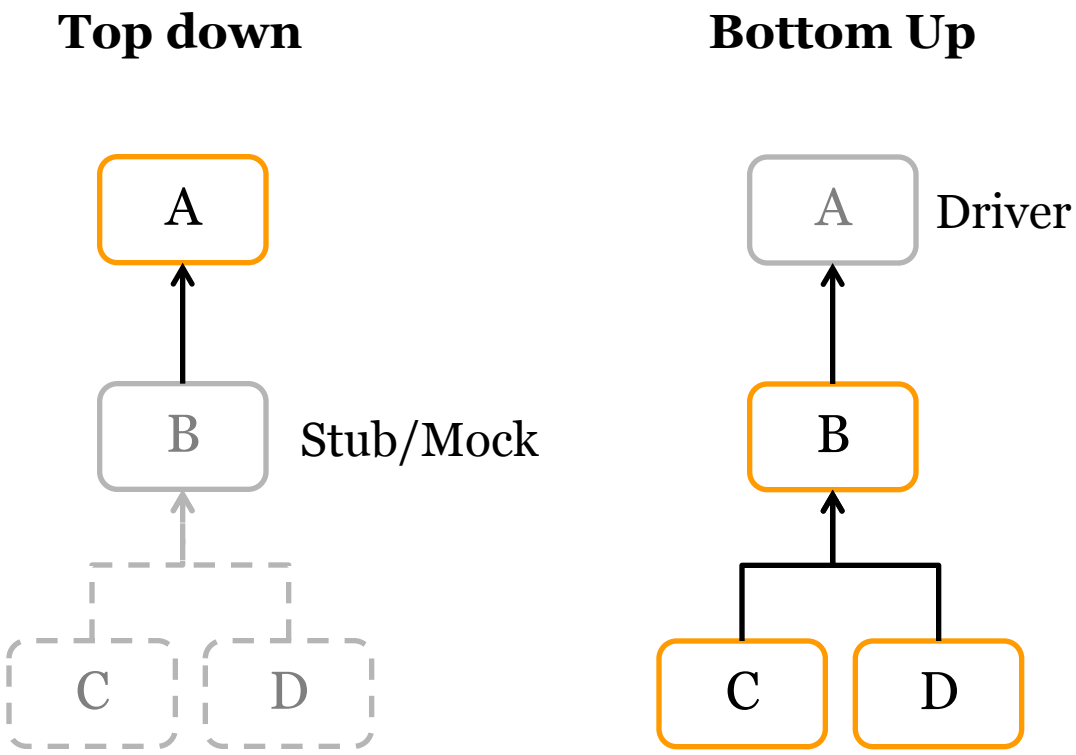
Driver - As stub but used in bottom up

Mock - A stub that contains code to verify the expected behavior

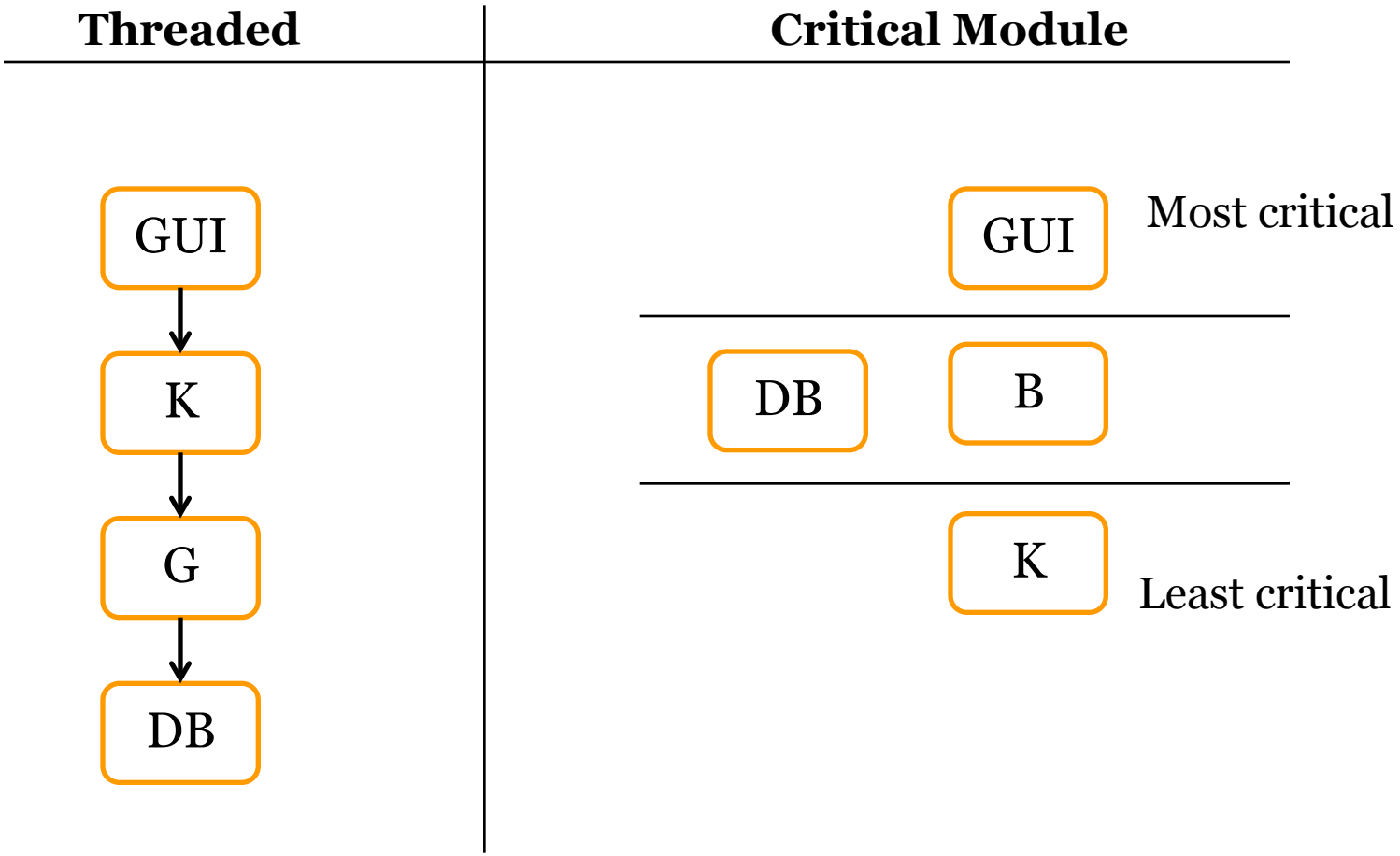
Sample strategies



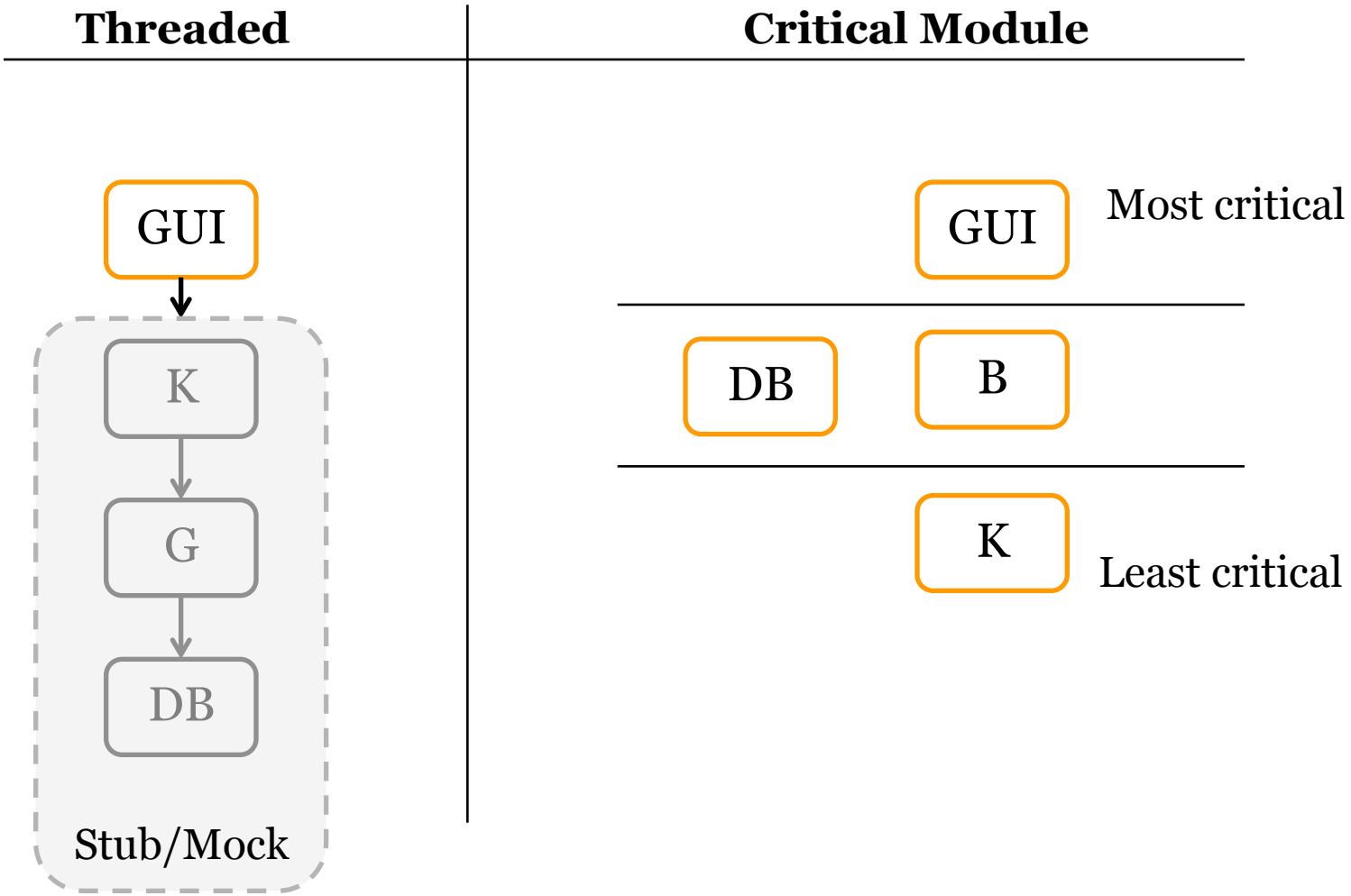
Structural oriented strategies



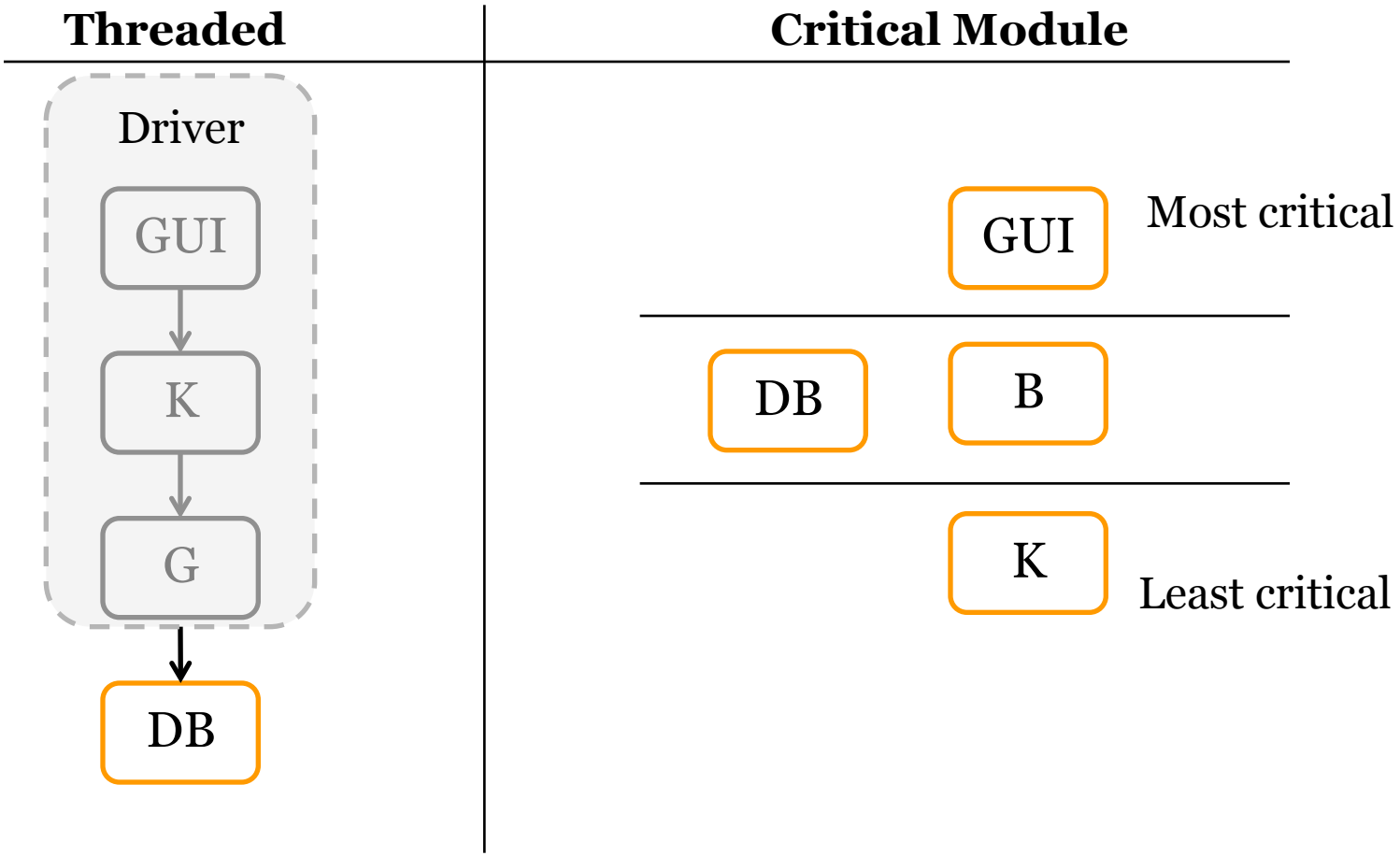
Feature oriented strategies



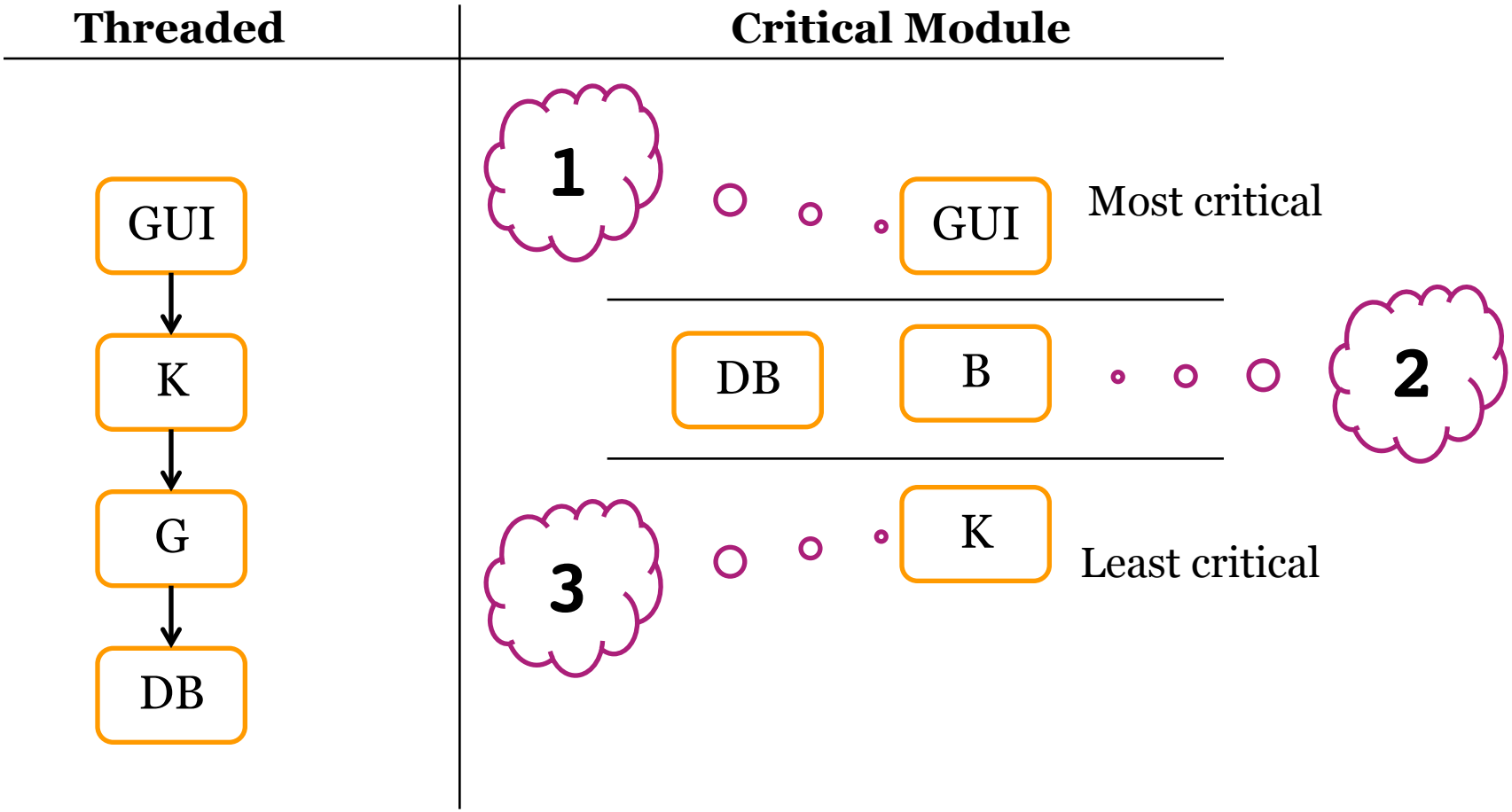
Feature oriented strategies



Feature oriented strategies



Feature oriented strategies



Continuous integration

NO COPYRIGHT ON THIS ONE ?

Picture source: http://blog.jki.net/news/niweek-2012-fire-and-forget-bulletproof-builds-using-continuous-integration-with-labview-video-slides-now-available/

Summary

- Avoid big bang integration
- Mostly more than one strategy is used
- Often mixed strategies

Further reading

- "**Software Testing and analysis**"
  by Mauro Pezzé & Michal Young
- https://www.ida.liu.se/~TDDD09/timetable/SystemAnatomy.pptx
- http://en.wikipedia.org/wiki/System_anatomy

PROMPT Software Testing

Integration Testing Strategies

SWEDISH ICT SICS

>PROMPT