

Problems with Monolith & SOA

Memi Lavi
www.memilavi.com



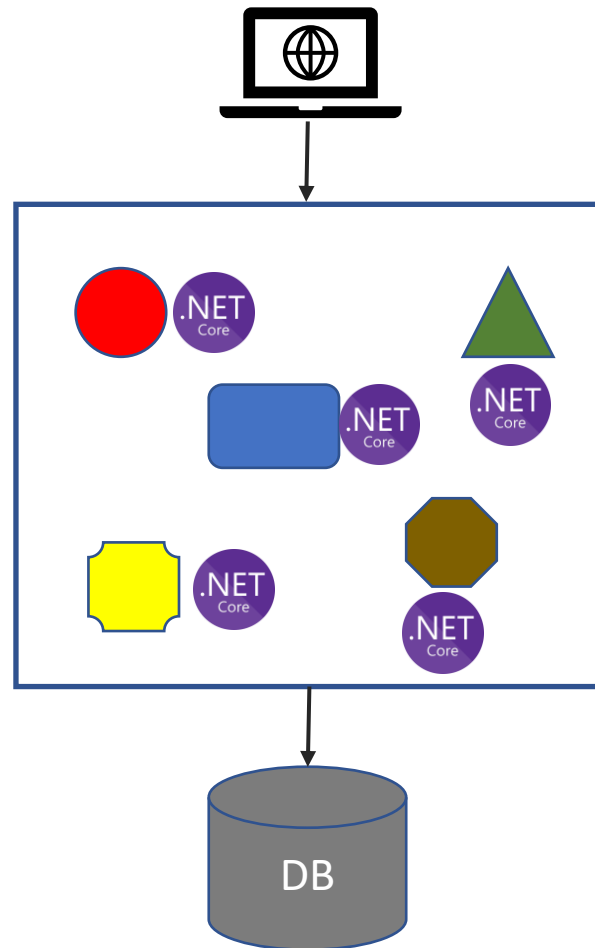
Problems

- A lot of problems were found in both paradigms
- Problems relevant to technology, deployment, cost and more
- We'll begin with the Monolith and then move on to SOA

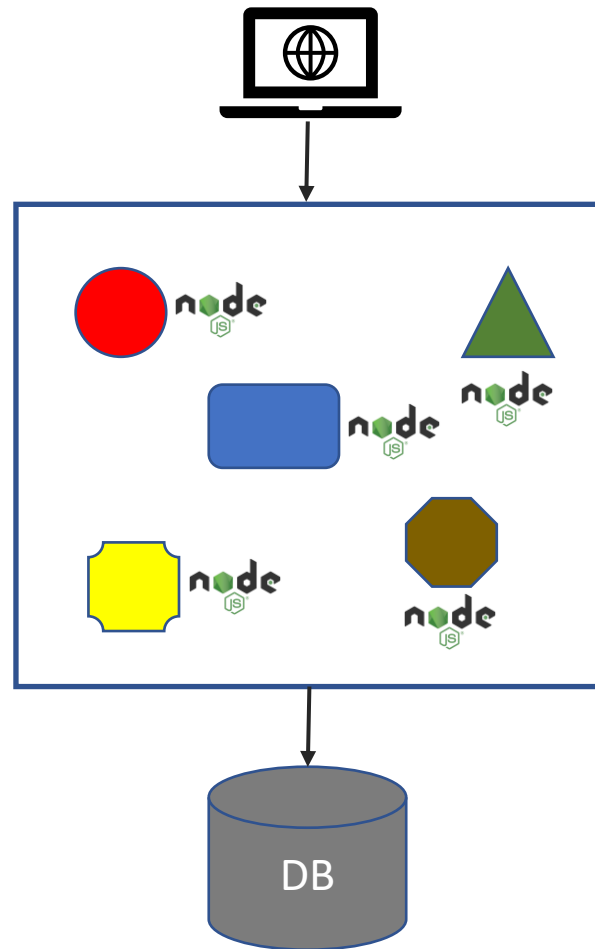
Single Technology Platform

- With monolith, all the components must be developed using the same development platform
- Not always the best for the task
- Can't use specific platform for specific features
- Future upgrade is a problem – need to upgrade the whole app

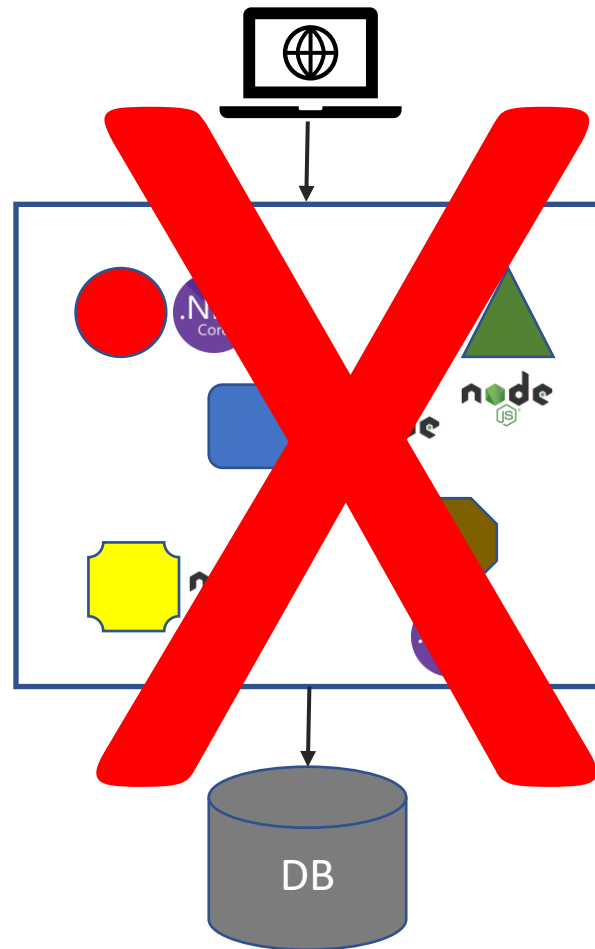
Single Technology Platform



Single Technology Platform



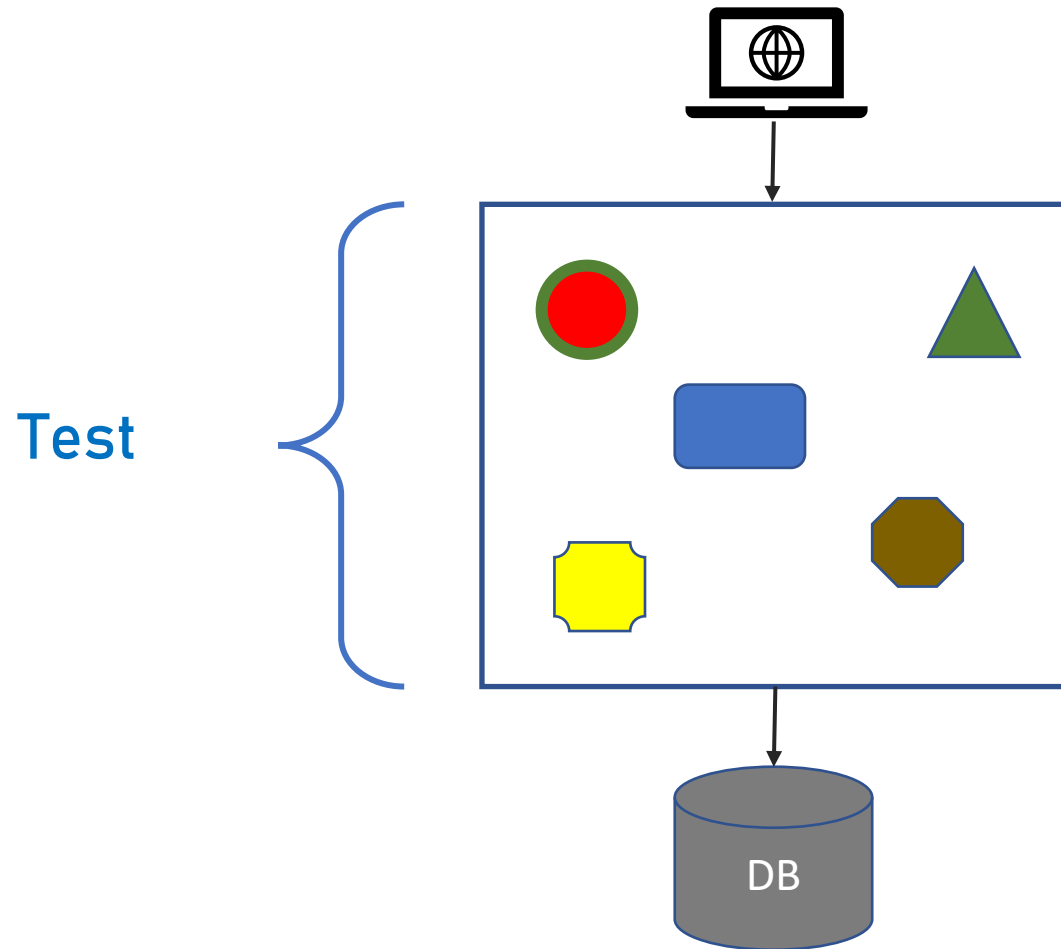
Single Technology Platform



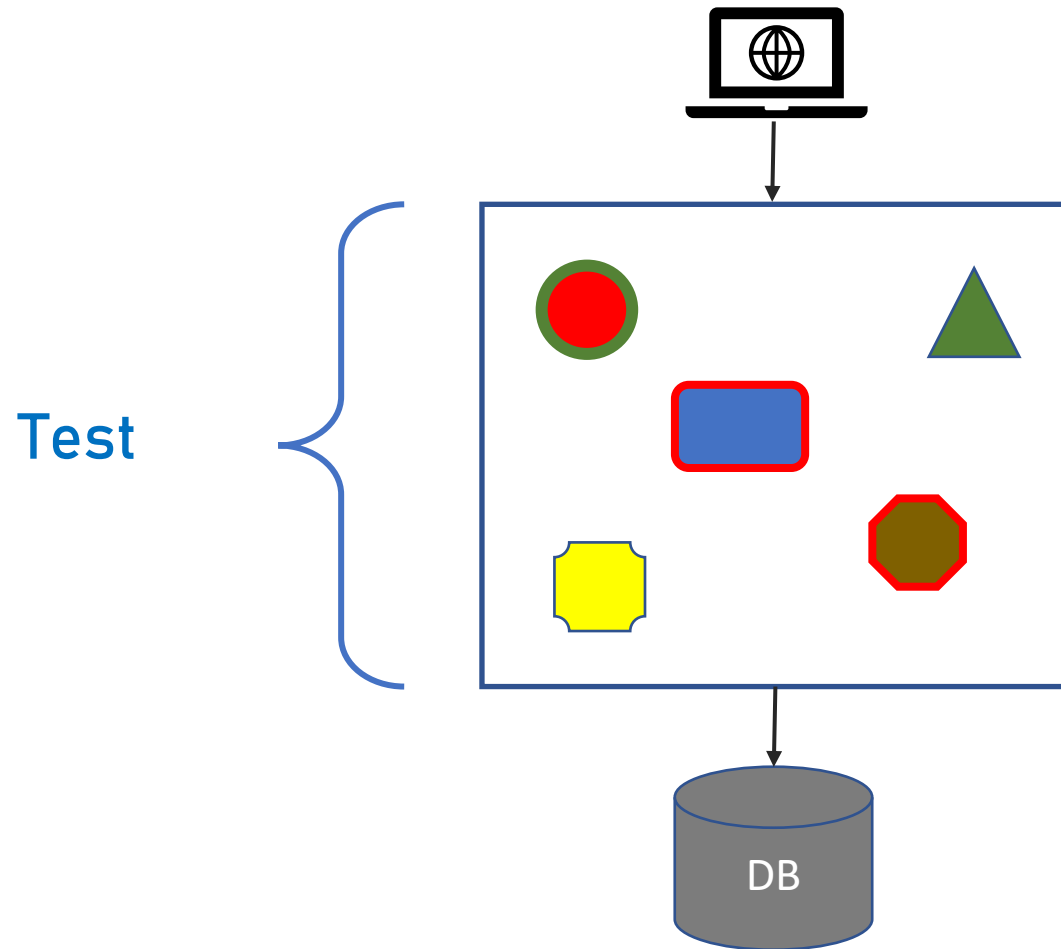
Inflexible Deployment

- With monolith, new deployment is always for the whole app
- No way to deploy only part of the app
- Even when updating only one component – the whole codebase is deployed
- Forces rigorous testing for every deployment
- Forces long development cycles

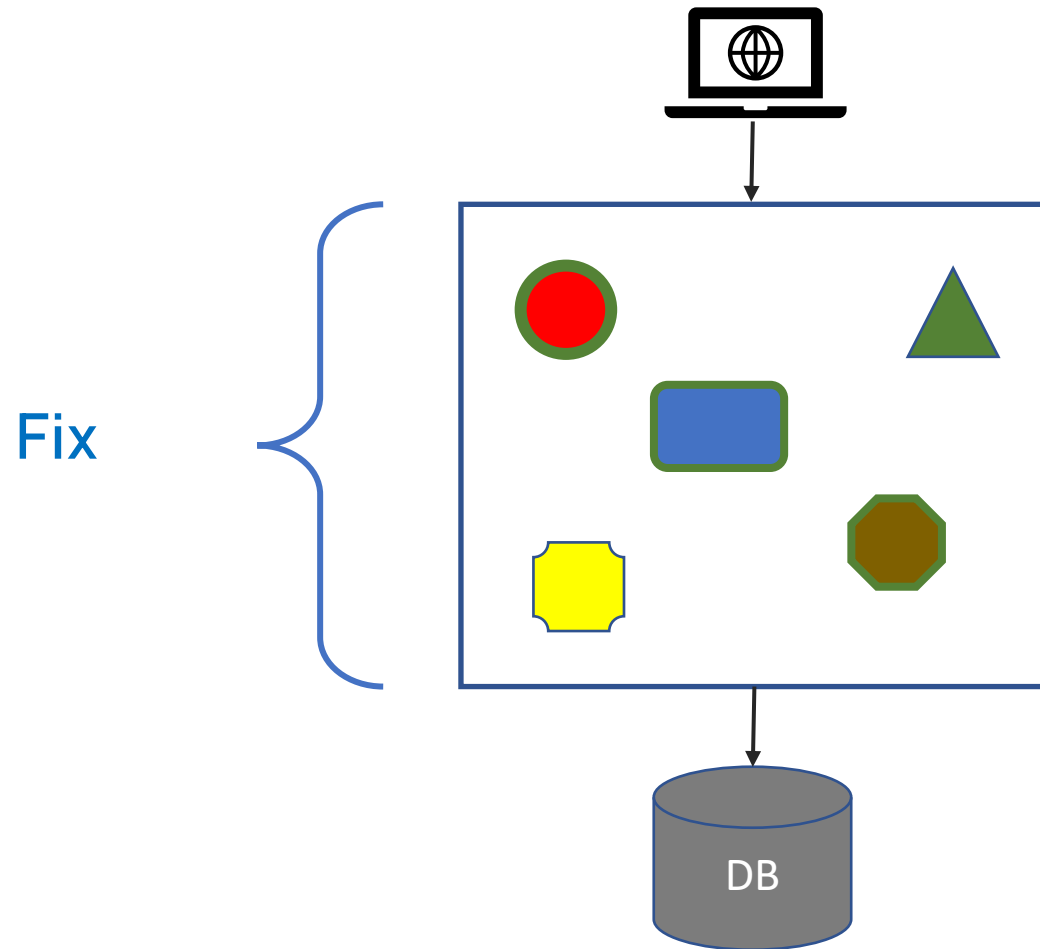
Inflexible Deployment



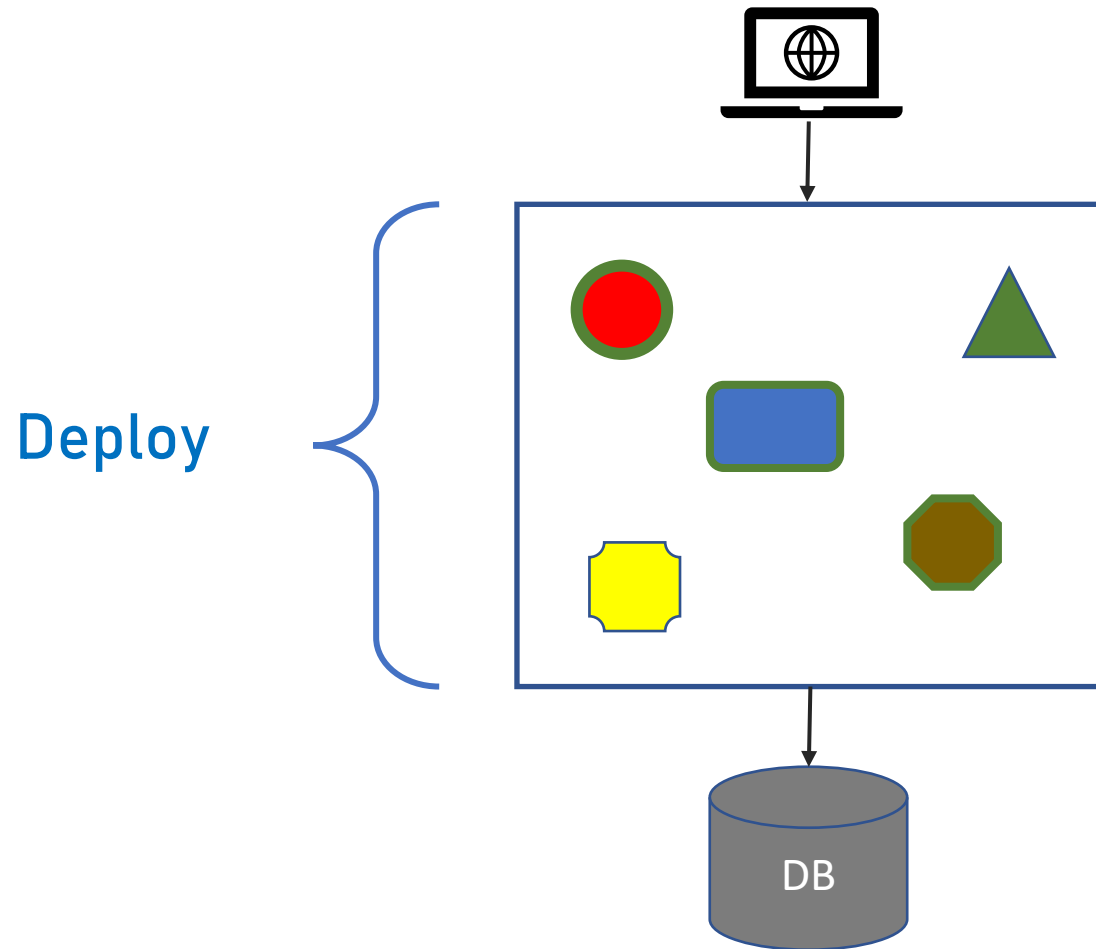
Inflexible Deployment



Inflexible Deployment



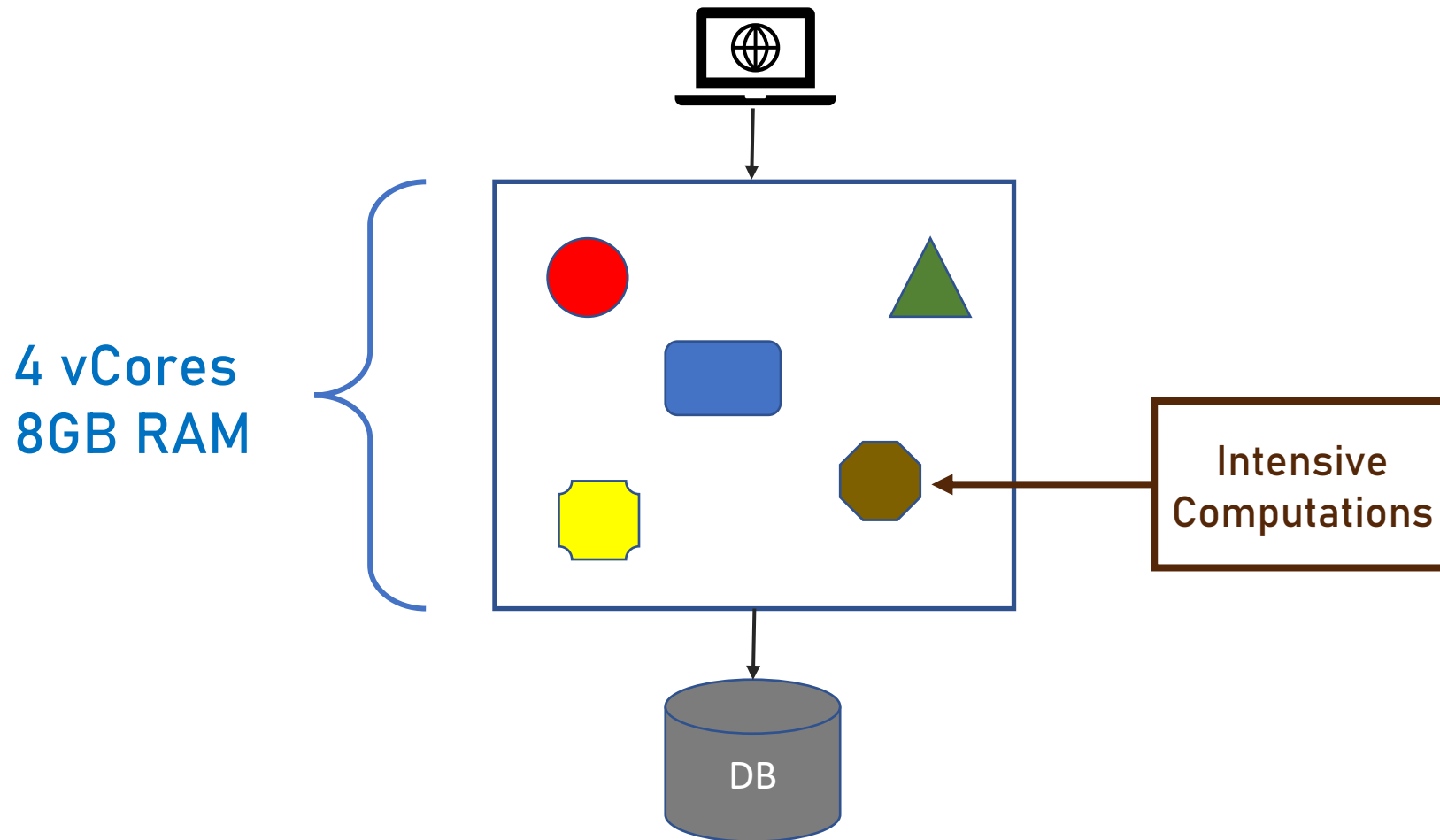
Inflexible Deployment



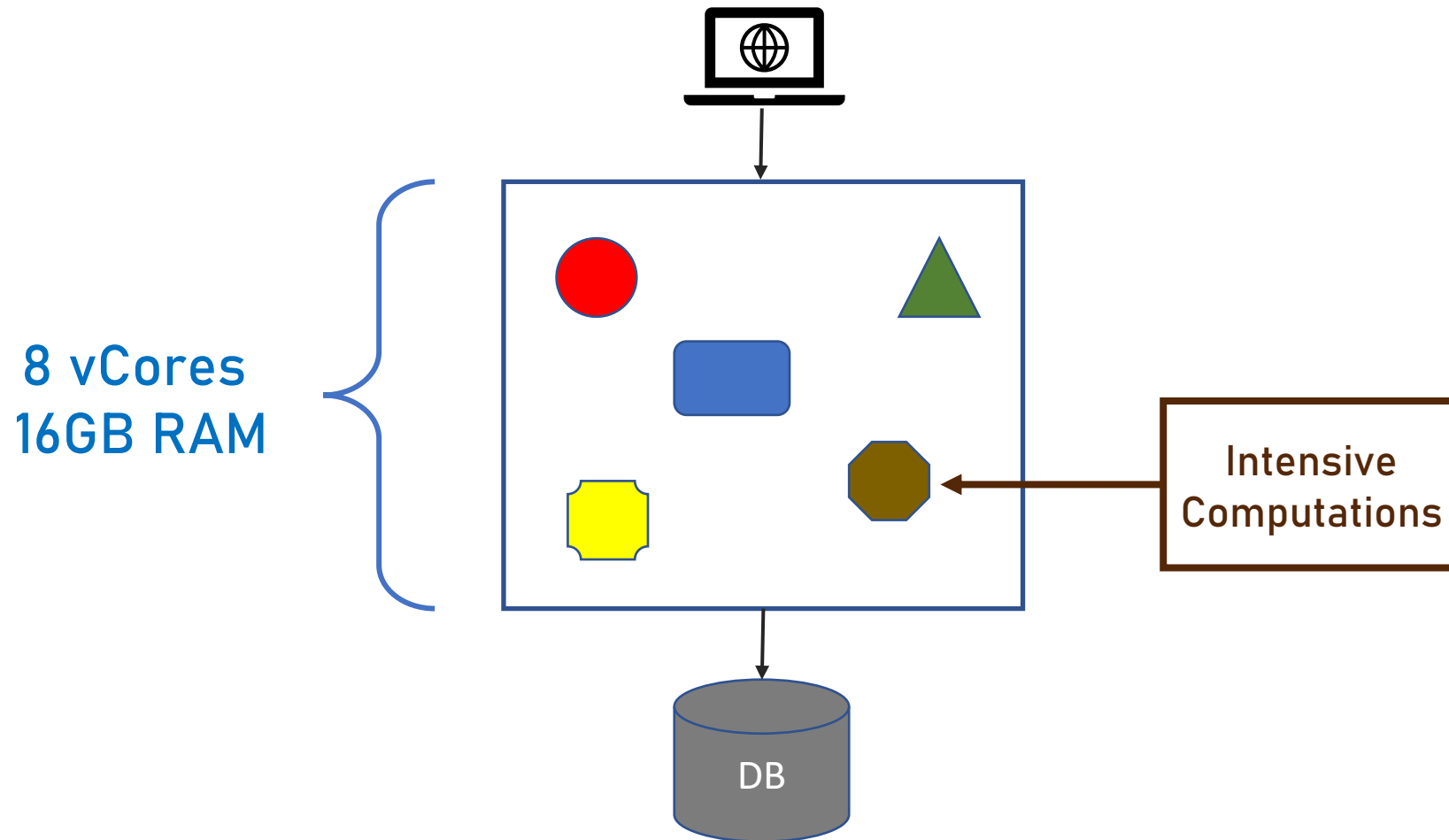
Inefficient Compute Resources

- With monolith, compute resources (CPU and RAM) are divided across all components
- If a specific component needs more resources – no way to do that
- Very inefficient

Inefficient Compute Resources



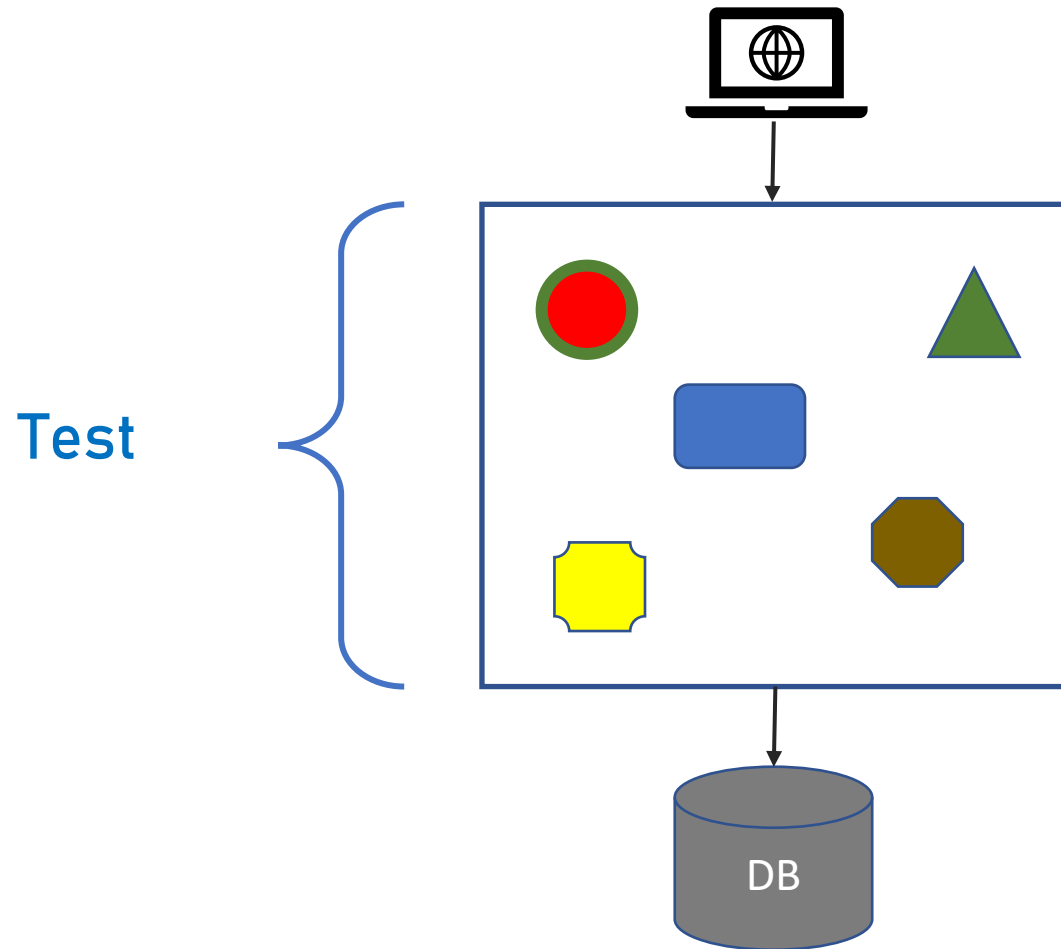
Inefficient Compute Resources



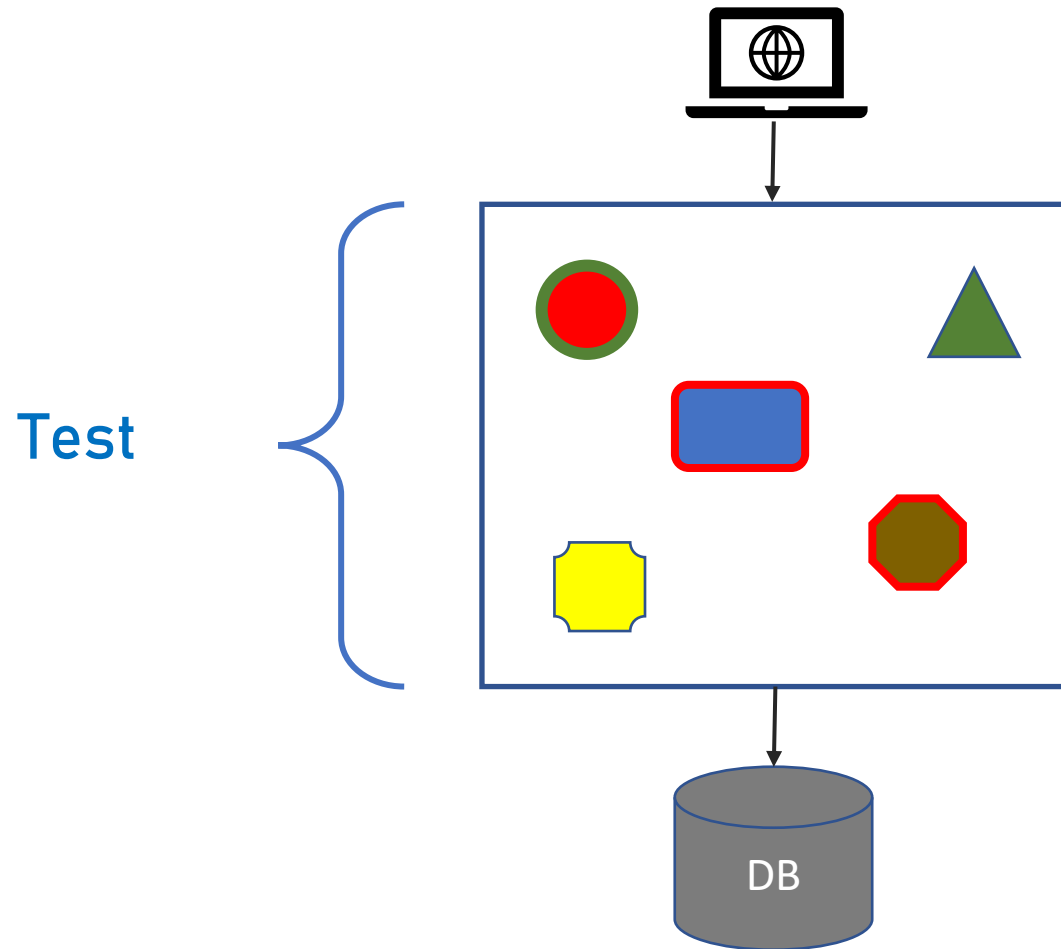
Large and Complex

- With monolith, the codebase is large and complex
- Every little change can affect other components
- Testing not always detects all the bugs
- Very difficult to maintain
- Might make the system obsolete

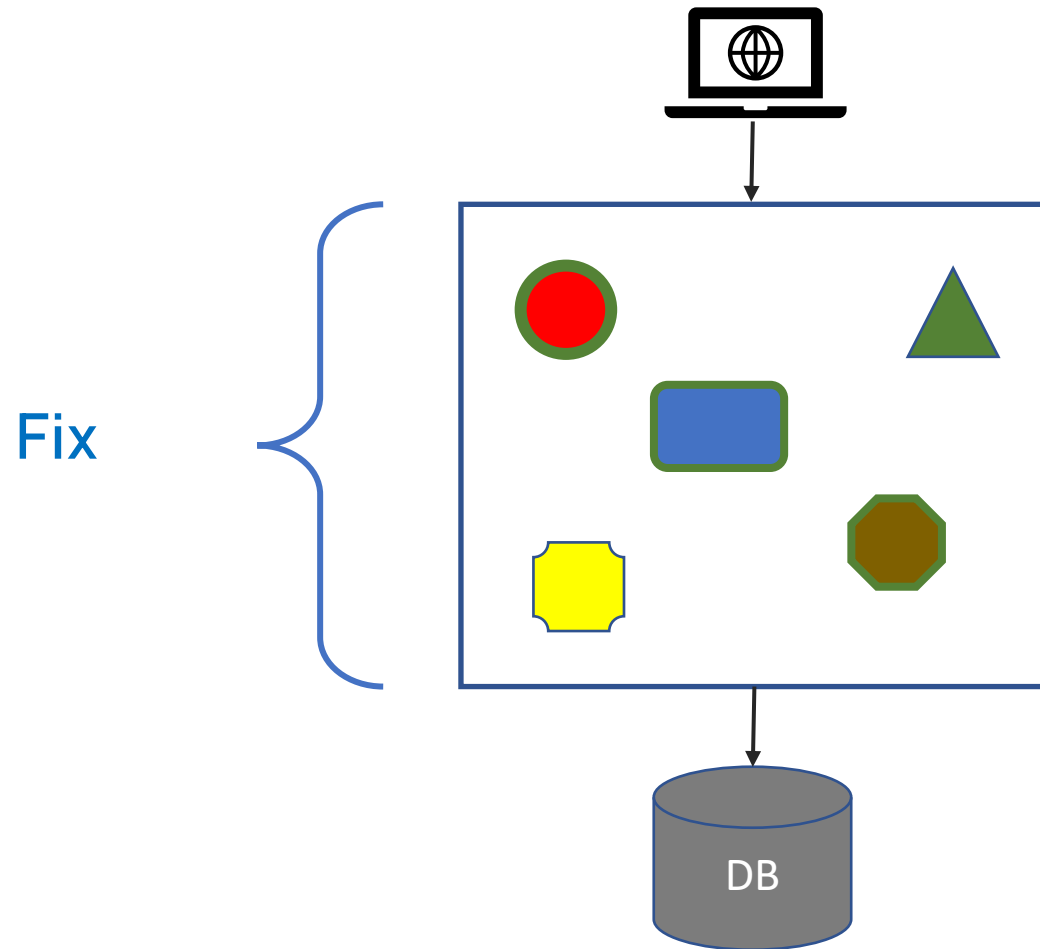
Large and Complex



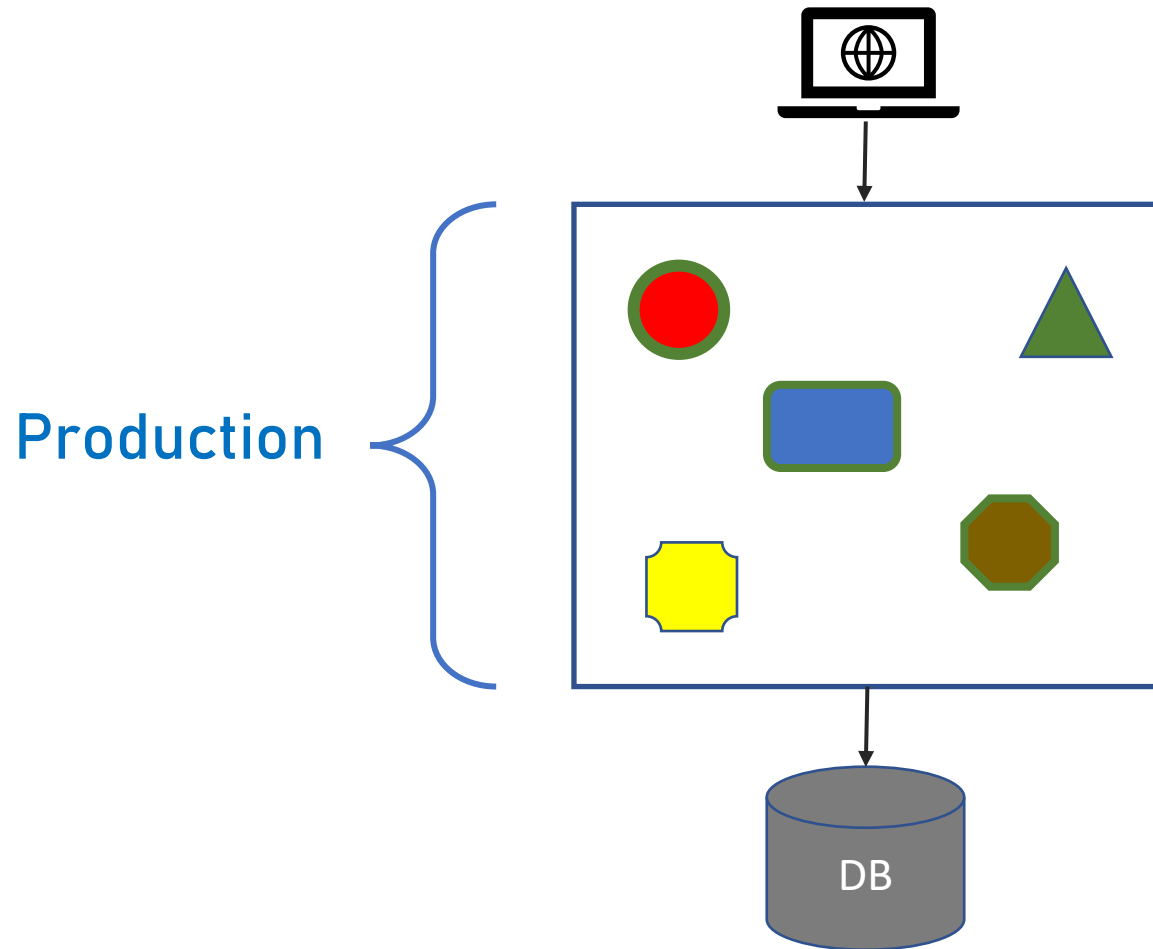
Large and Complex



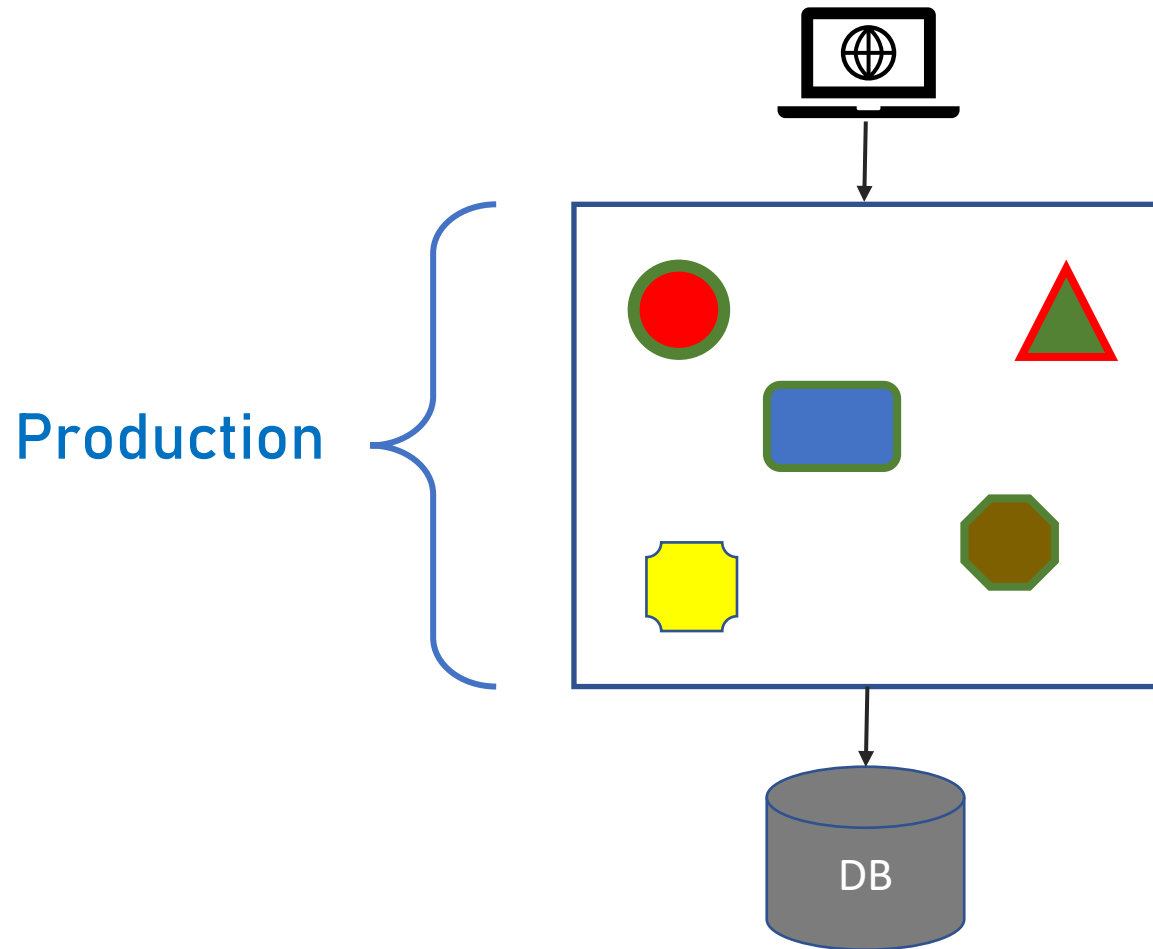
Large and Complex



Large and Complex



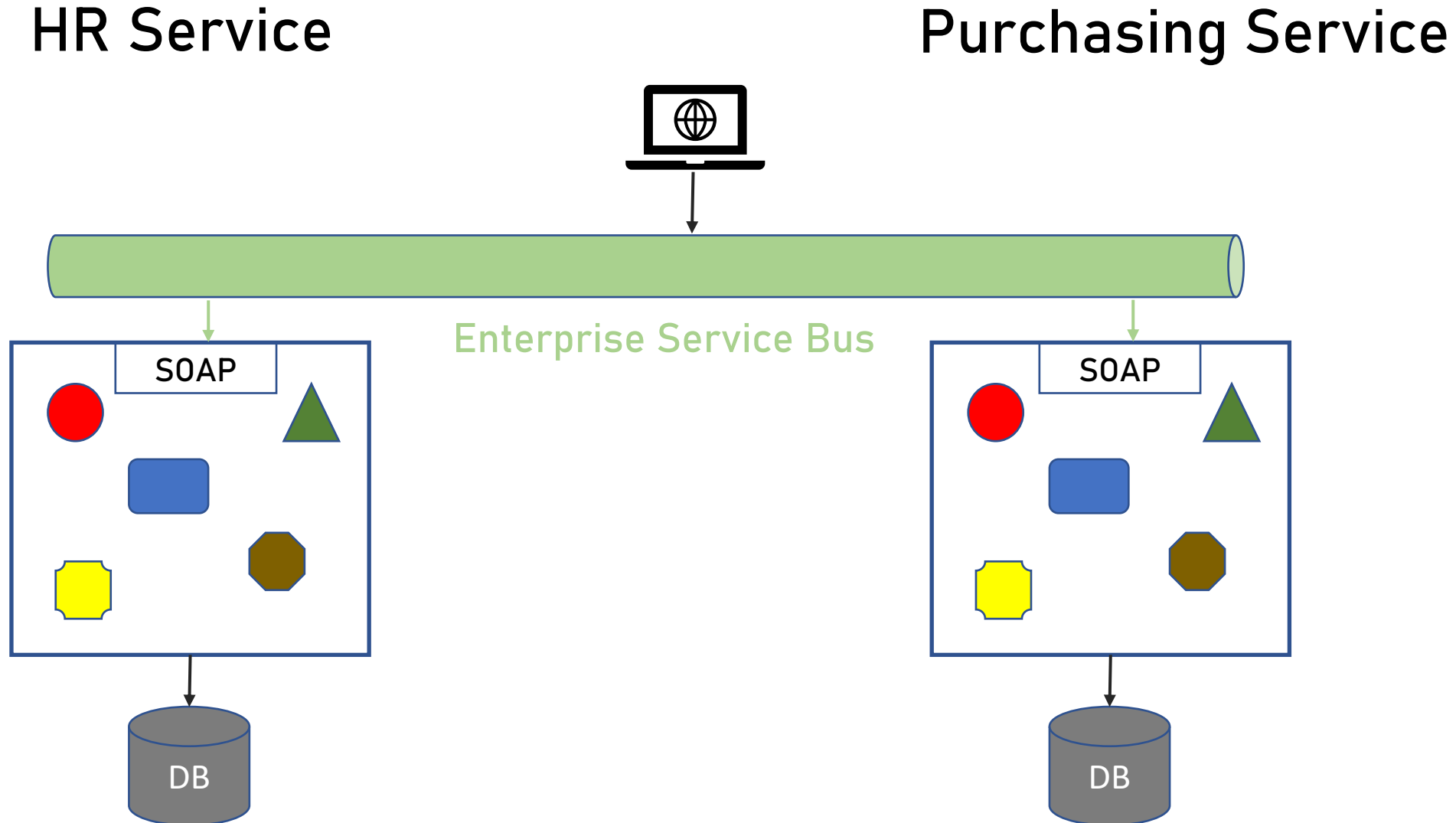
Large and Complex



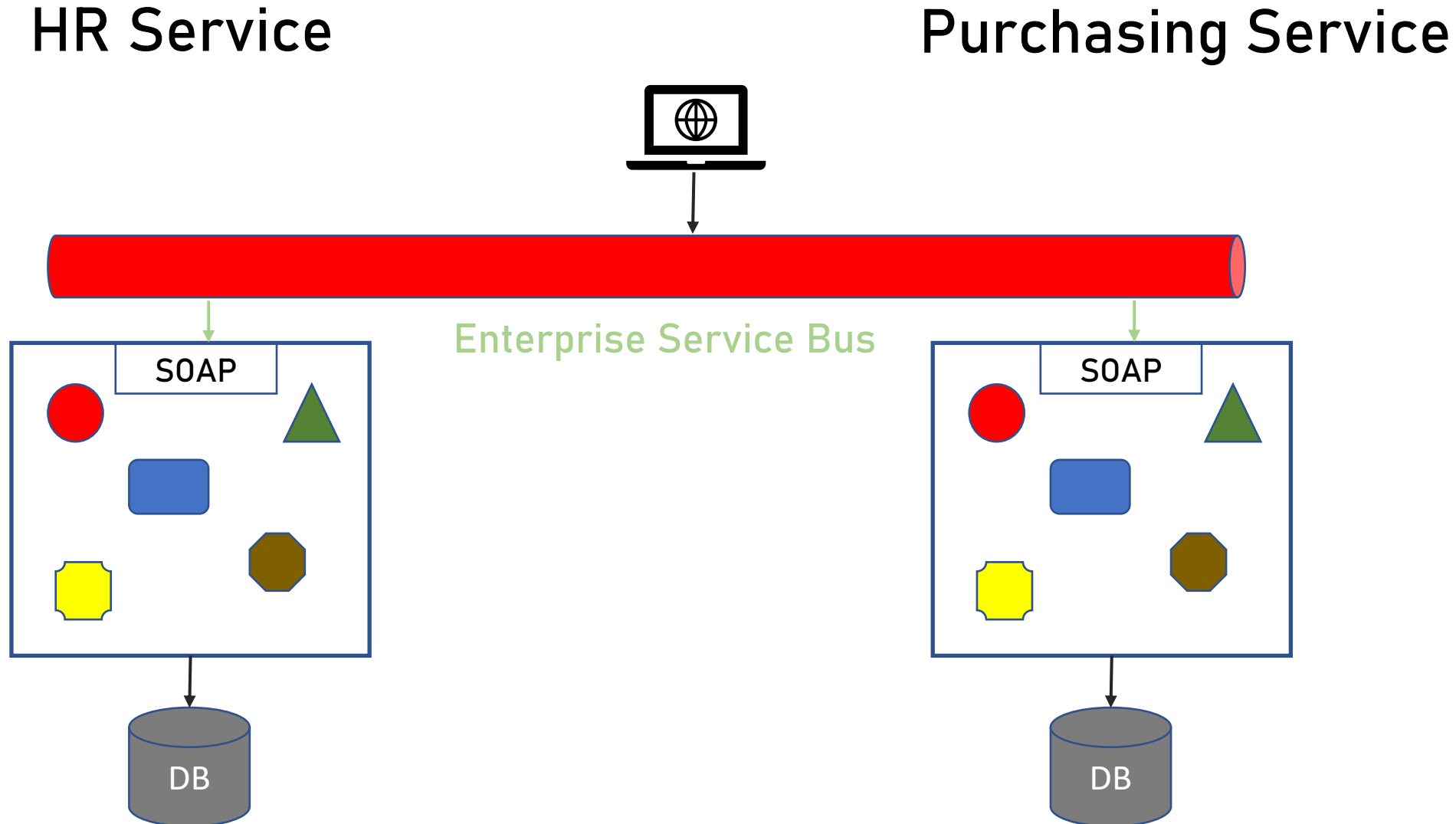
Complicated and Expensive ESB

- With SOA, the ESB is one of the main components
- Can quickly become bloated and expensive
- Tries to do everything
- Very difficult to maintain

Complicated and Expensive ESB



Complicated and Expensive ESB



Lack of Tooling

- For SOA to be effective, short development cycles were needed
- Allow for quick testing and deployment
- No tooling existed to support this
- No time saving was achieved