

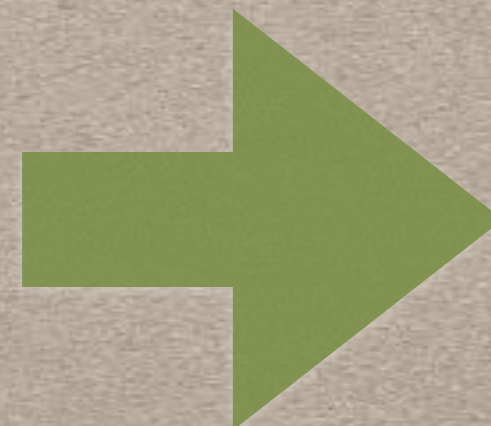


# SIDECAR PATTERN

JAMES GOUGH  
@JIM\_\_GOUGH



# BUILDING MICROSERVICES





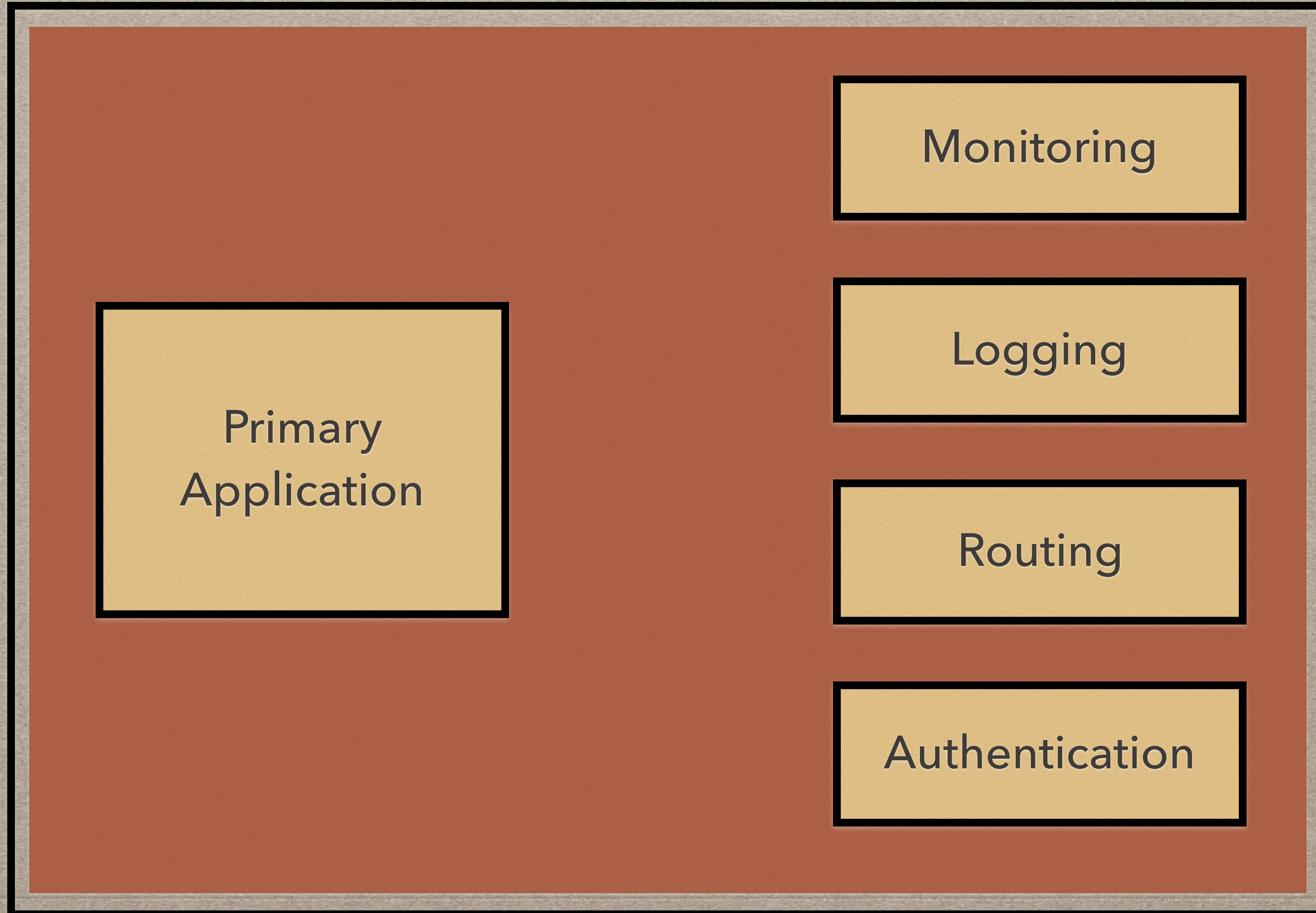
# MICROSERVICE CHALLENGES

- Microservices are not micro when they include a huge stack of libraries
- Communication between services can be tricky
- Can be much harder to test
- Need to have a good automated builds



# SIDECAR PATTERN

*Container*





# ADVANTAGES OF SIDECAR

- Independent Runtime Environment
- Has access to the same resources as main application
- Reduces the latency when communicating
- Can be used with different languages
- Can share functionality with other teams easily



# CONSIDERATIONS

- The sidecar pattern is well suited to containers
- Carefully consider interprocess communication
- Have to balance library usage with sidecar
- Increases resource costs
- If the service needs to scale at a different rate



# SIDECAR PATTERN

JAMES GOUGH  
@JIM\_\_GOUGH

