Stateless and Stateful EDA

Memi Lavi
www.memilavi.com

Stateless vs Stateful EDA

- There are two main patters in implementing EDA
- Stateless and Stateful
- Related to the consumers behavior
- Both are legitimate, but make sure to select the right one for your scenario

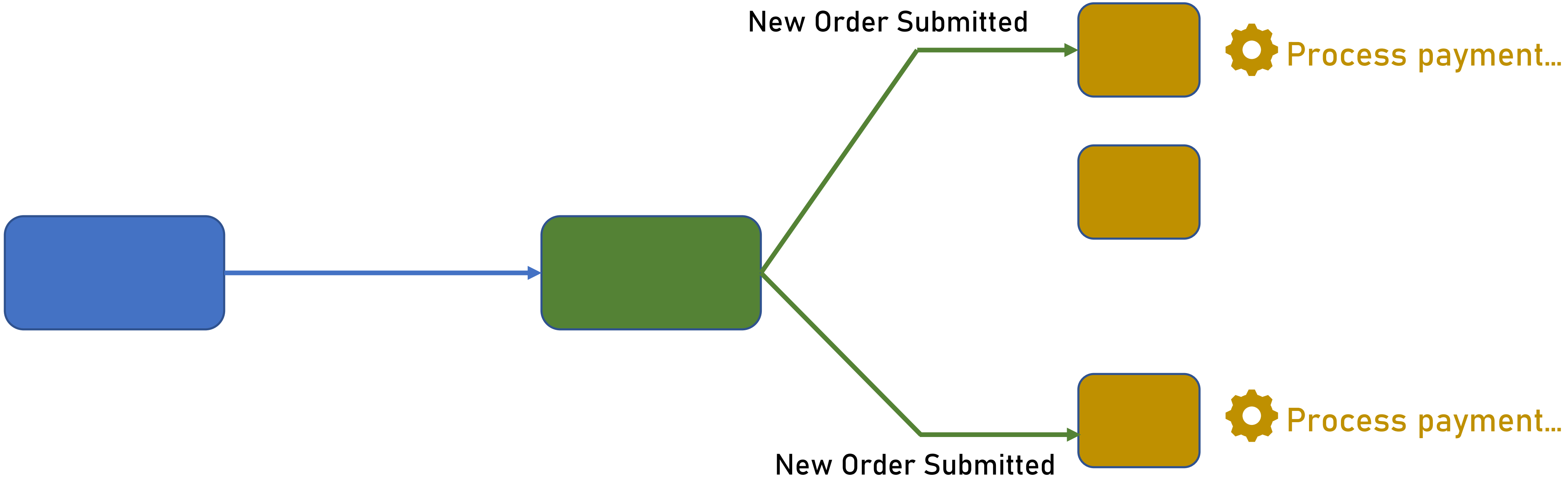
Stateless vs Stateful EDA

- In software architecture there’s also the stateless vs stateful debate
- While the concepts are similar, the reasoning is different
- With software architecture it’s often said that:
  - “Stateful is bad”
- This is not necessarily the case with EDA

Stateless EDA

- Each event handled by a consumer is completely autonomous and is not related to past / future events
- Should be used when the event is an independent unit with its own outcomes

Stateless EDA



Payment processing consumers

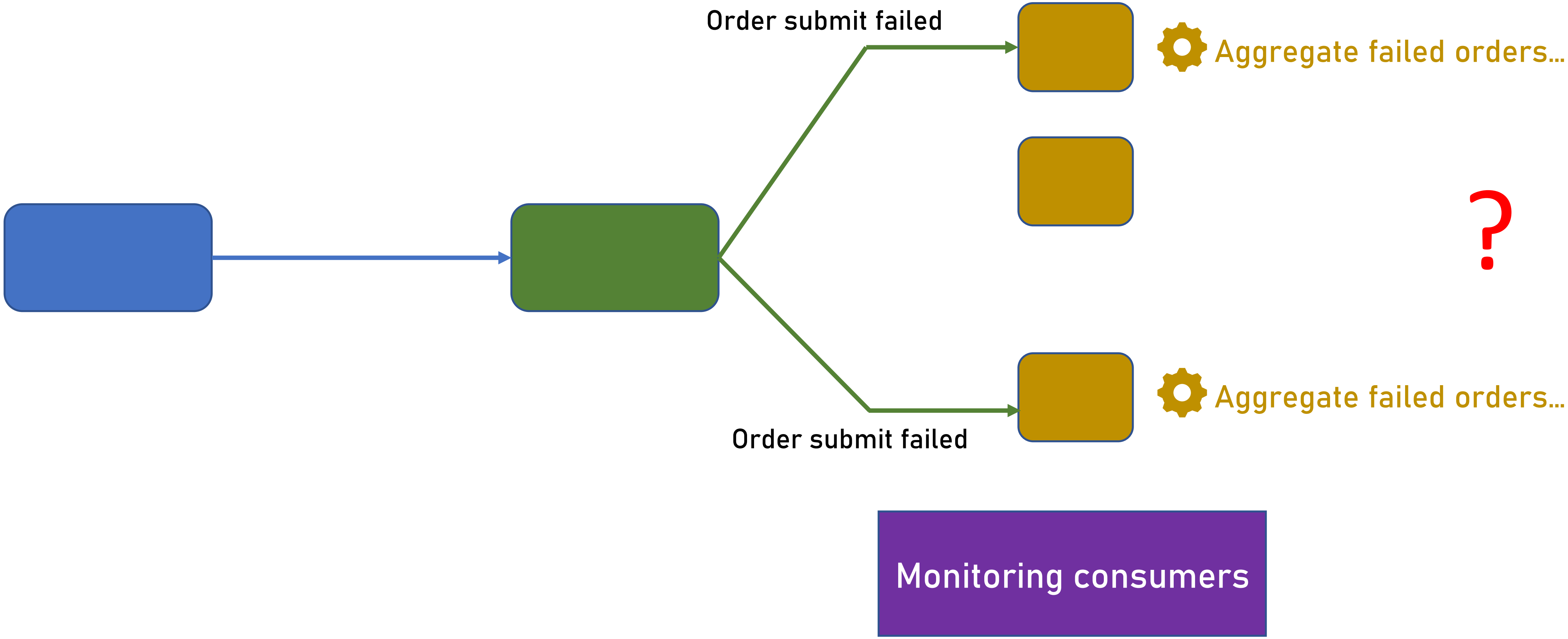
Stateless EDA

- It doesn’t matter which consumer is handling the event
- The outcome is always the same
- Should be used when each event is autonomous
- Note: Has nothing to do with the question of what data is contained in the event and whether a call to a DB is required

Stateful EDA

- Events might be related to past / future events
- Should be used mainly for aggregators and time-related events
- Examples:
  - Send an email if more than 5 failure events were received in a single minute
  - Calculate the amount of orders submitted in an hour

Stateful EDA



Stateful EDA

- It's extremely important which consumer handles the event
- Current state is stored in specific consumer(s)
- Should be used when events are part of a chain of events

Problems with Stateful EDA

- Stateful EDA presents some problems that should be taken care of

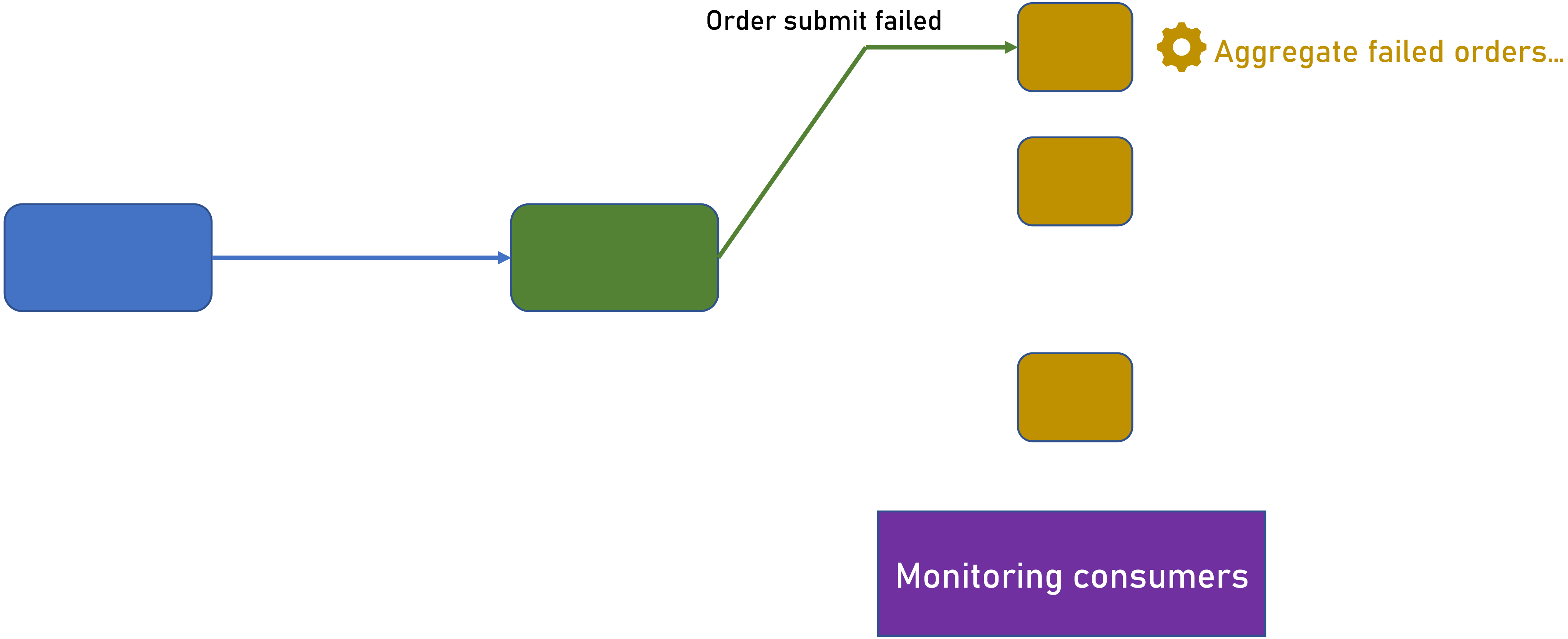
Load balancing

Scalability

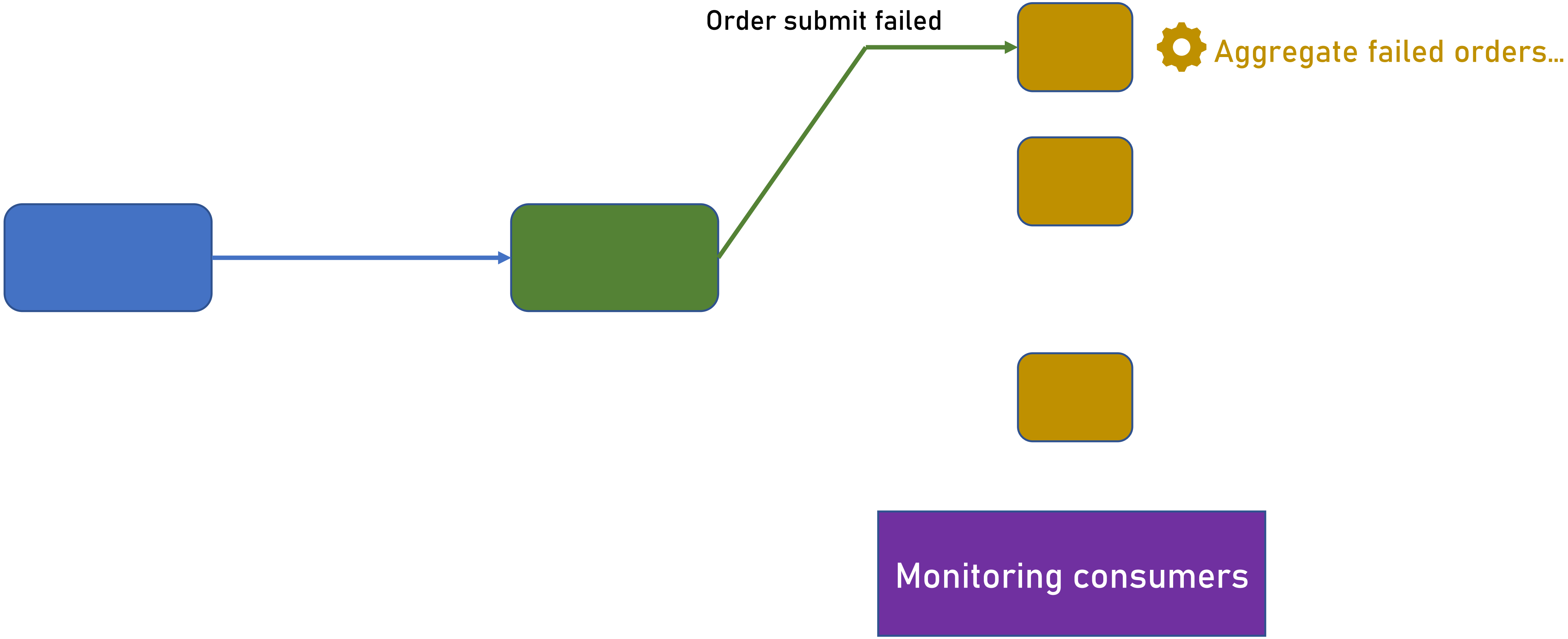
Load Balancing

- Since the state is stored in a specific consumer, subsequent events must be routed to the same consumer
- No load balancing is possible

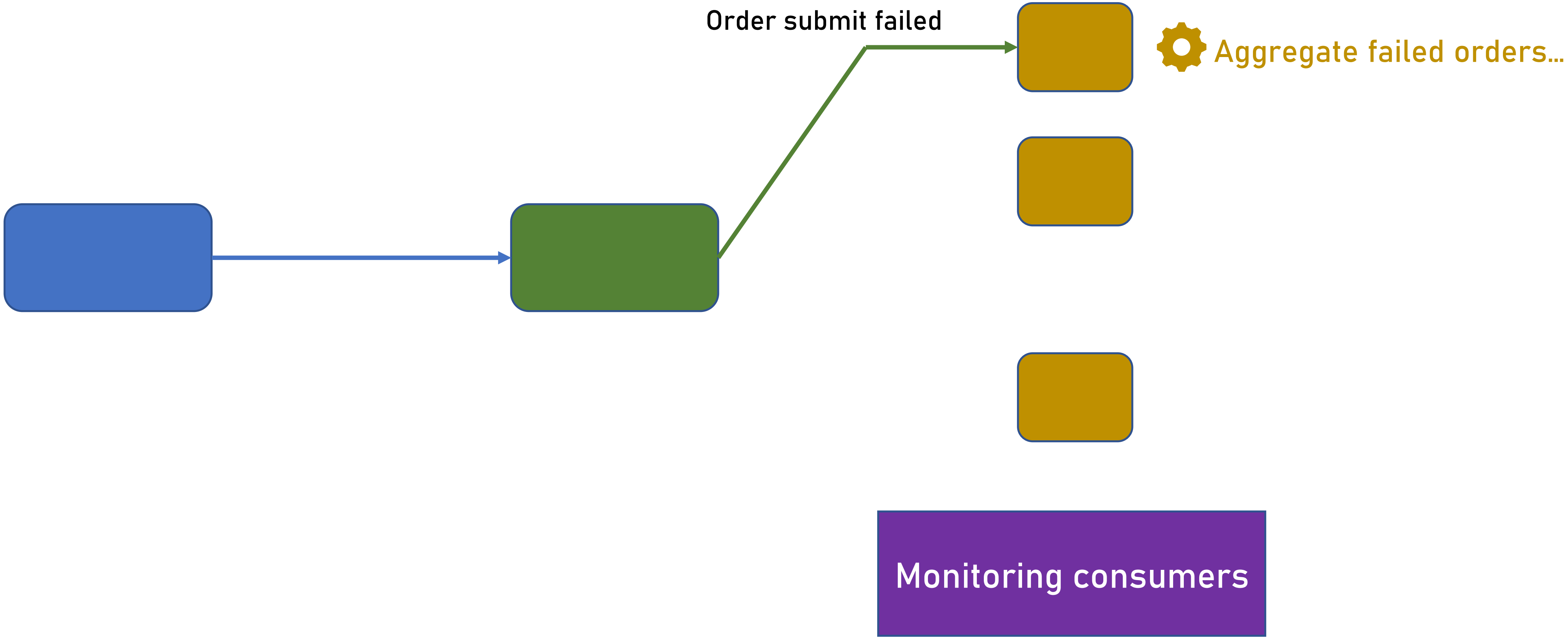
Load Balancing



Load Balancing



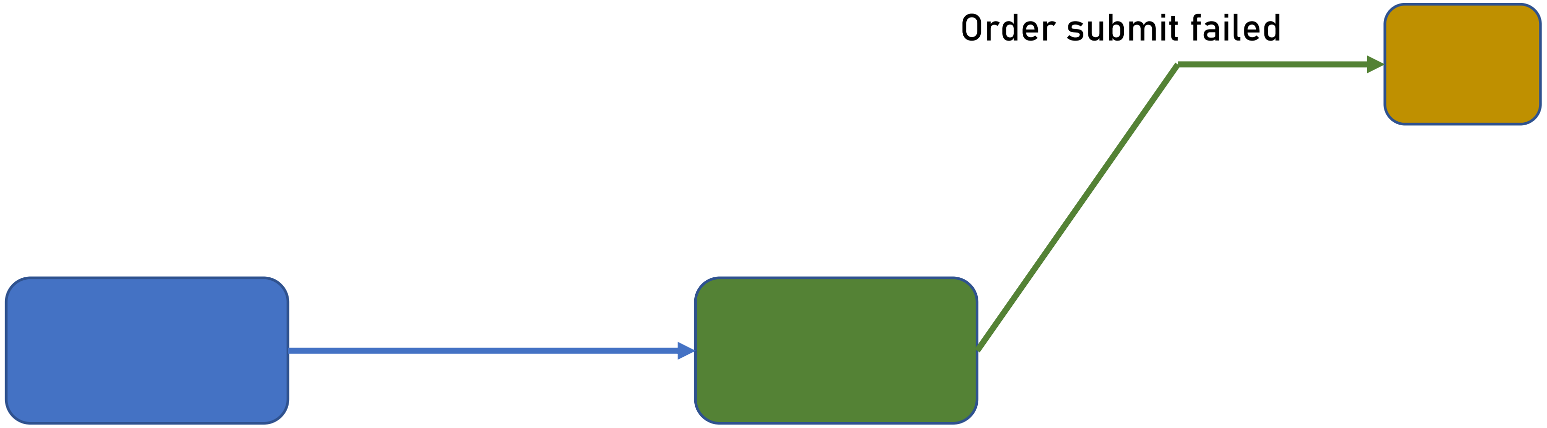
Load Balancing



Scalability

- Since the state is stored in a specific consumer, additional consumers cannot be added to handle the events

Scalability

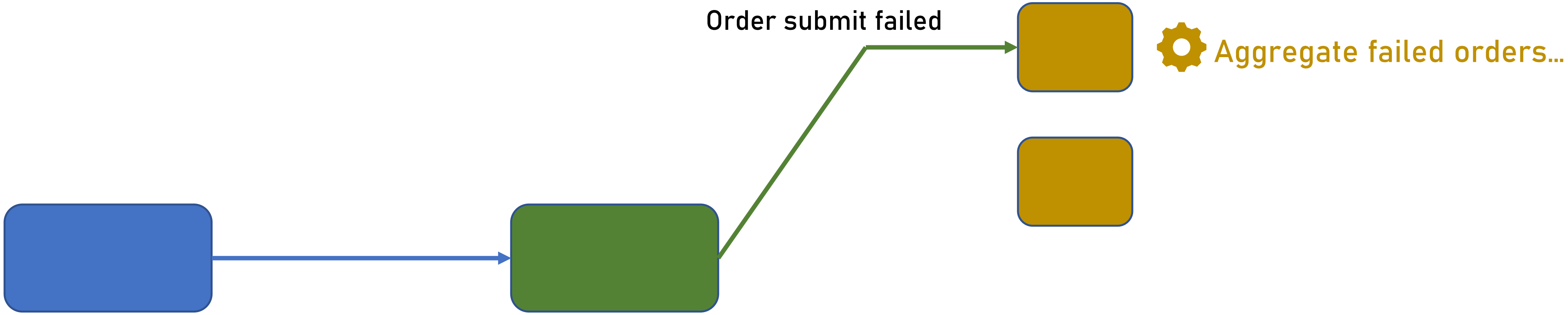


Order submit failed

Aggregate failed orders…

Monitoring consumers

Scalability



Monitoring consumers

Stateless vs Stateful

- Rule of thumb:
  - Use stateless EDA unless the business requirements force you to use stateful