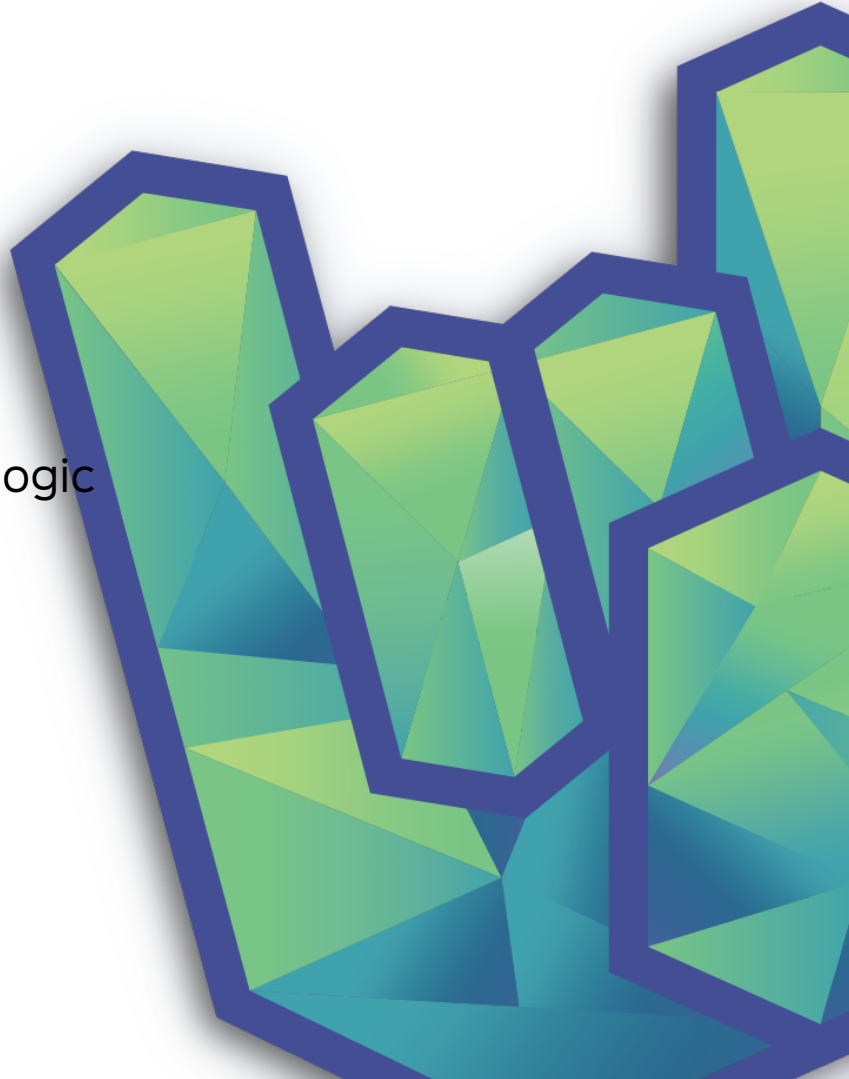


# Custom Operators



# Goal

Create our own components with custom logic  
Master the GraphStage API



# Input port methods

InHandlers interact with the upstream

- `onPush`
- `onUpstreamFinish`
- `onUpstreamFailure`

Input ports can check and retrieve elements

- `pull`: signal demand
- `grab`: take an element
- `cancel`: tell upstream to stop
- `isAvailable`
- `hasBeenPulled`
- `isClosed`

# Output port methods

OutHandlers interact with downstream

- onPull
- onDownstreamFinish

(no onDownstreamFailure as I'll receive a cancel signal)

Output ports can send elements

- push: send an element
- complete: finish the stream
- fail
- isAvailable
- isClosed

# Recap

## Custom components with GraphStage

```
class MyComponent extends GraphStage[MyShape] {  
  // component-specific members go here  
  val input = Inlet[Int]  
  override val shape = SourceShape(input)  
  
  override def createLogic(attributes: Attributes) = new GraphStageLogic(shape) {  
    // any mutable state goes here  
  
    setHandler(input, new InHandler {  
      override def onPush() {  
        // your logic here  
        pull(input)  
      }  
    })  
  }  
}
```

→ Handler callbacks *never called concurrently*

DO NOT expose mutable state outside these handlers!

**Akka rocks**

