

Command



Karoly Nyisztor

DEVELOPER

@knyisztor www.leakka.com

Overview

Motivation

Animation commands demo

- Encapsulate animation method invocations in command objects

Enhanced command

- Embed references to receiver and arguments in the command protocol

Undoable and macro commands

- Add support for undo operations and macro commands

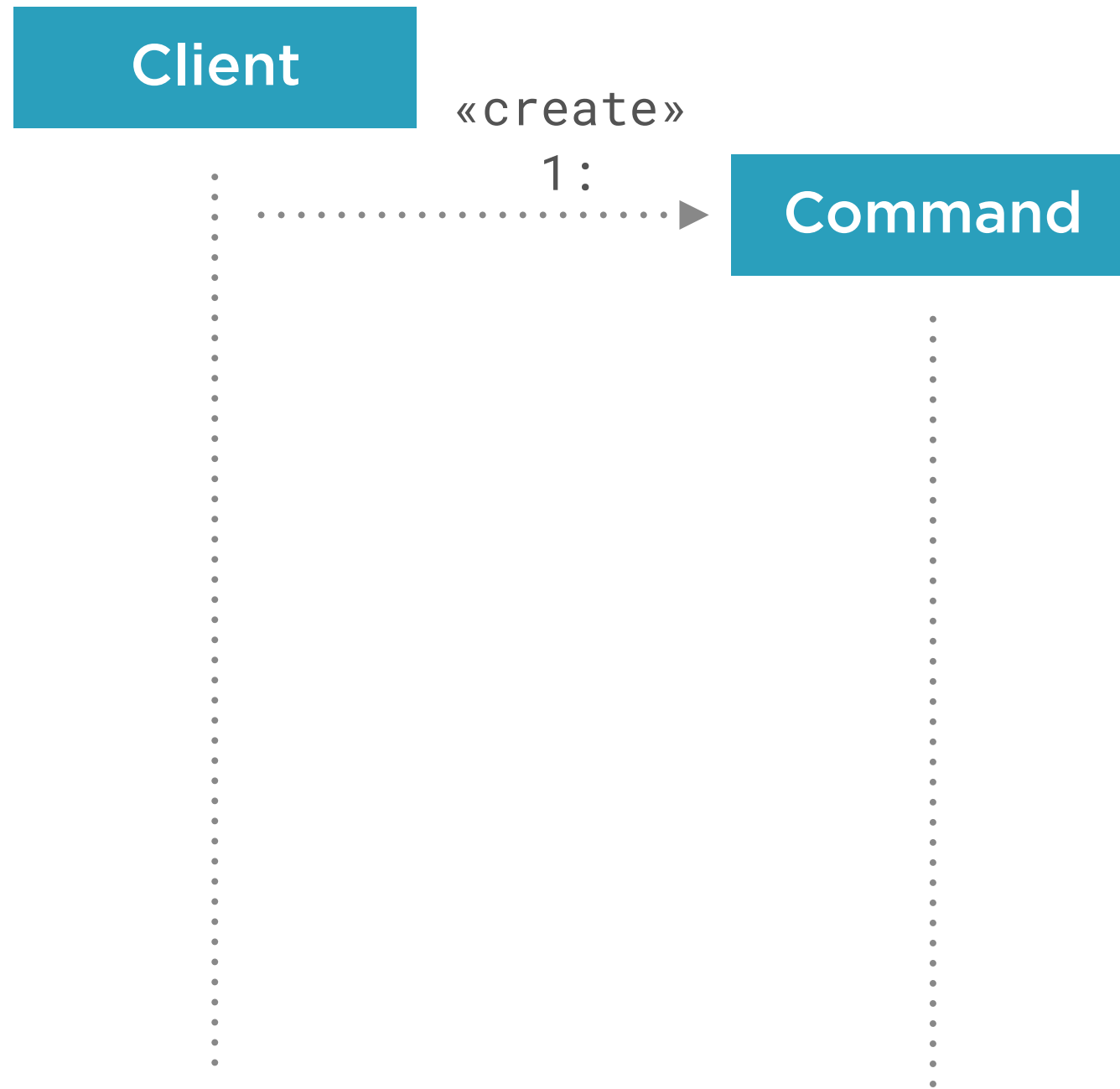
Motivation



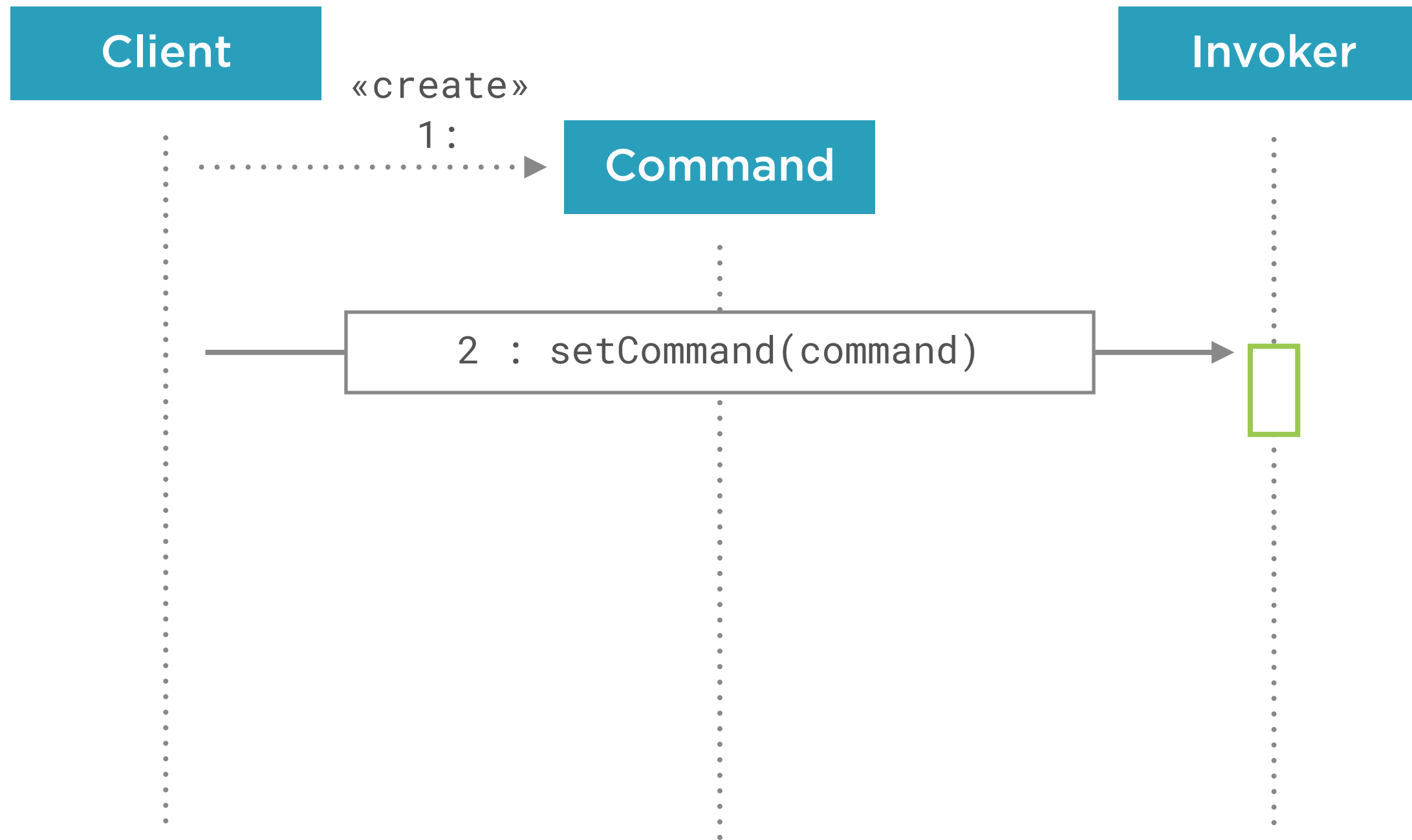
Encapsulate method invocation

- Package method invocation and receiver in a command object
- Decouple request invoker from receiver
- Support for undoable operations
- Create macro commands

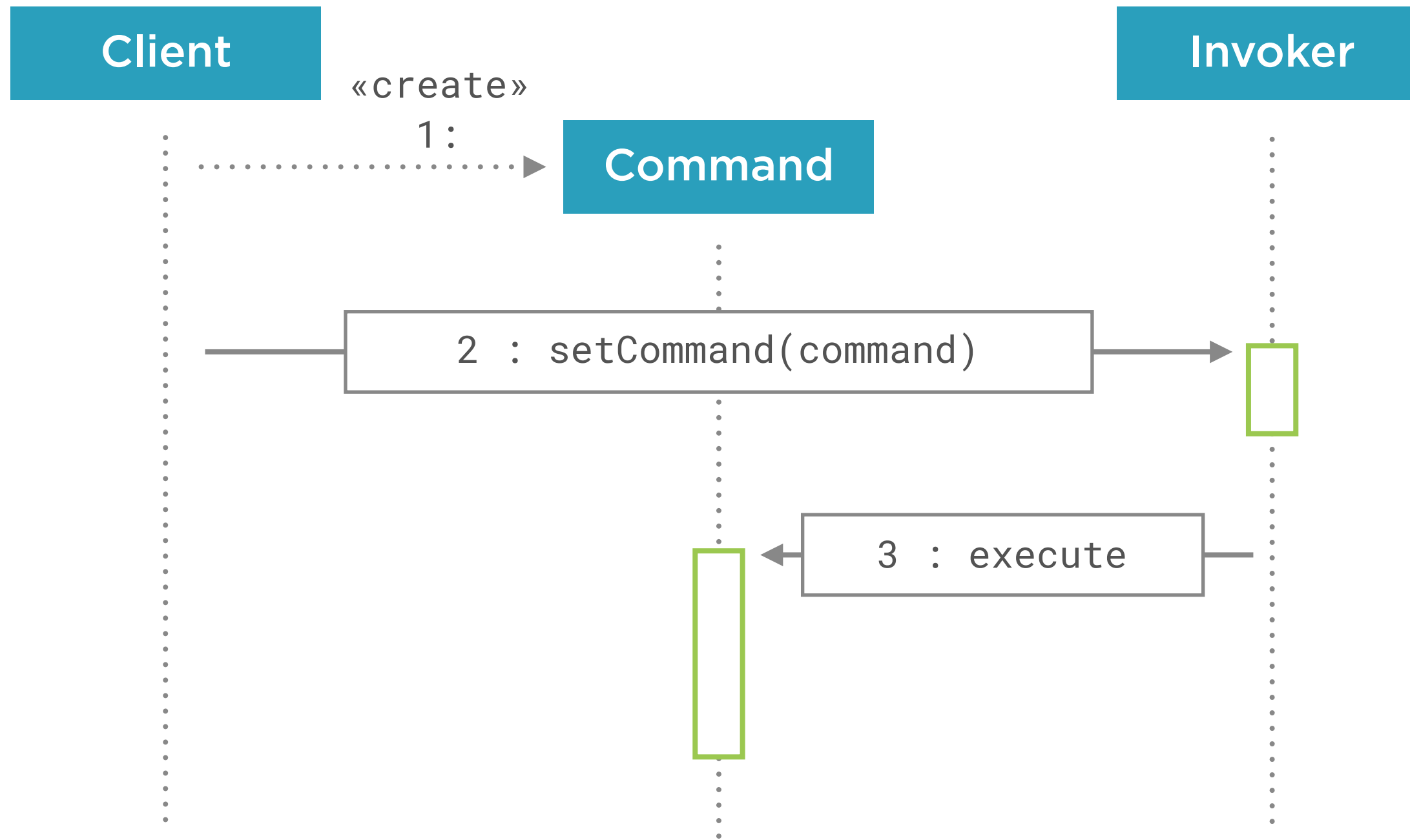
Creates Command



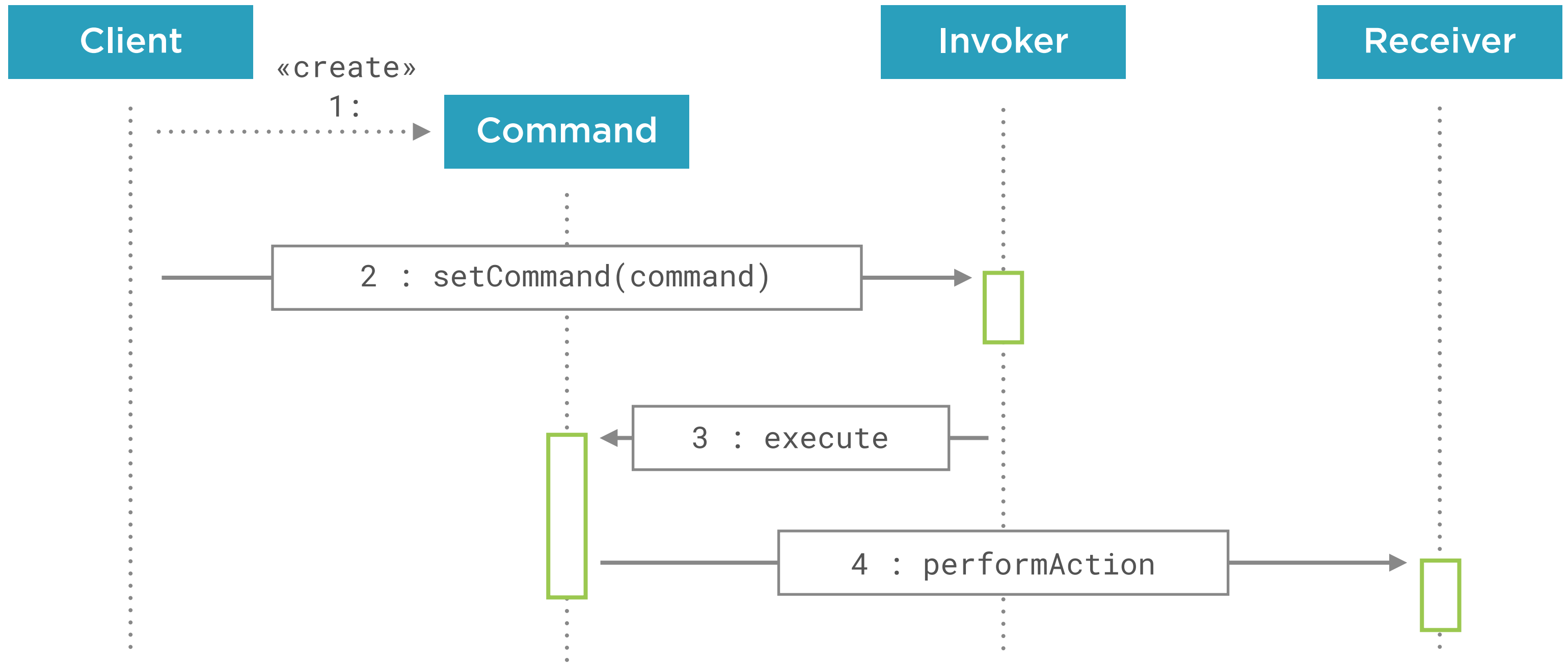
Set Command on Invoker



Invoke Command Object's execute() Method



Action Performed by Receiver



Command

Encapsulates a request into an object. Allows performing requests without knowing the actions or the receiver.

Demo

Animation commands demo

- Encapsulate animation method invocations in command objects

Demo

Enhanced command

- Embed references to receiver and arguments in the command protocol

Demo

Undoable and macro commands

- Add support for undo operations and macro commands

Summary

The Command design pattern:

- Encapsulates method invocations
- Allows invoking actions without knowing the actions or the receiver
- Decouples the requester from the object which performs the action

Pitfalls

- Exposing details of the receiver object or the request