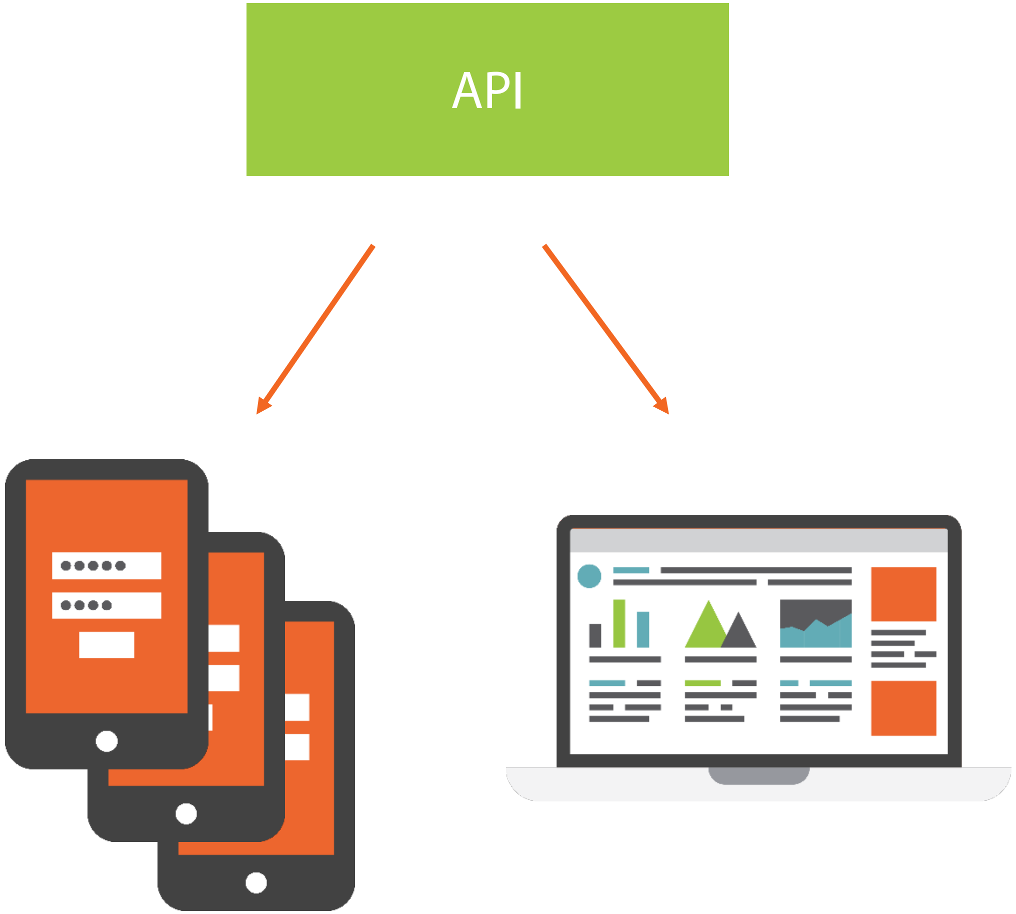
Building and Securing a RESTful API for Multiple Clients in ASP.NET

Introduction

Kevin Dockx

@KevinDockx | http://blog.kevindockx.com/



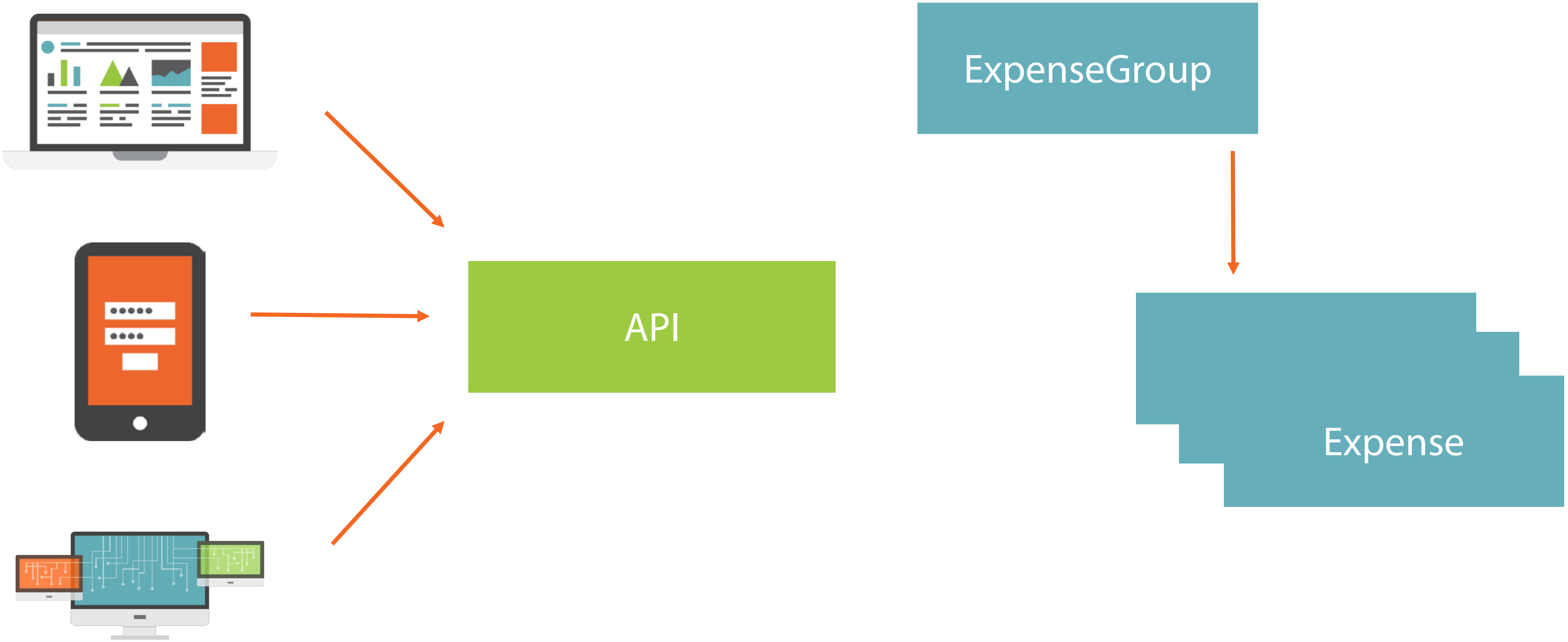
- API is bound by technical limitations
- Different clients impose different functional requirements on our API
- The type of communication changes how we consume the API
- The client imposes restrictions on where the API has to live
- That in turn leads us to a certain way to secure it

So What Can You Expect?

- You'll learn how to build a REST-ful API (with Web API)
- … that can be consumed by a variety of (cross-platform) clients
- … realistically deployed
- … secured in a way that works cross-platform, with OAuth2 & OpenID Connect

- … ALWAYS starting from the requirements

Say Hello to the Expense Tracker



API

ExpenseGroup

Expense

Gathering Requirements

Technical

Functional

Environmental

- API needs to be consumable from different types of clients (standards!)
- API needs to be friendly to consume (uniform interface)
- API needs to support CRUD operations
- API needs to support sorting
- API needs to support paging
- API needs to support data shaping for associations

Gathering Requirements

Technical

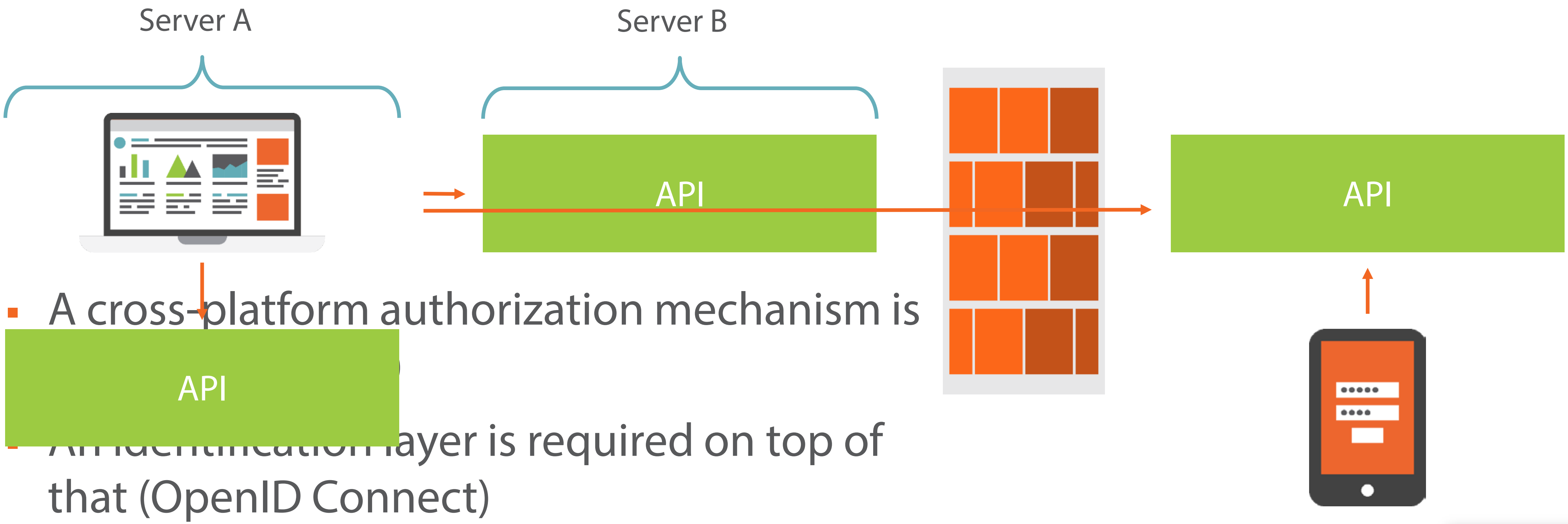
Functional

Environmental

- A cross-platform authorization mechanism is required (OAuth2)
- An identification layer is required on top of that (OpenID Connect)
- API needs to support filtering
- API needs to support data shaping for field-level selection
- API mustn't break when changes are rolled out

Gathering Requirements

Technical | Functional | Environmental



- A cross-platform authorization mechanism is)
- An identification layer is required on top of that (OpenID Connect)

Summary

API requirements are defined by technical limitations, functional requirements and environmental variables

We'll learn how to build a REST-ful API that fits these for multiple (cross-platform) clients

We'll learn how to secure it