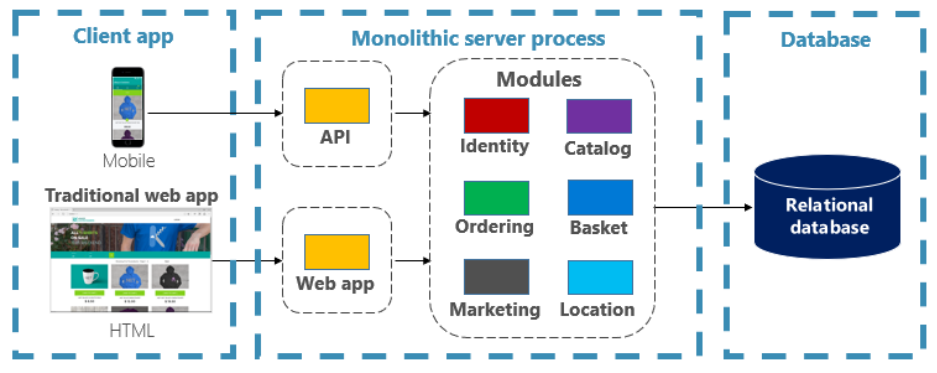
Architecting Cloud-Native .NET Apps for Azure

# Introduction to cloud-native applications

#11

**Problem: We want to design a major e-Commerce application.**

Monolithic Solution:



What we have done is we construct a large core application containing all of domain logic, it includes modules such as Identity, Catalogue, Ordering and more. The core app communicates with a large relational database. The core exposes functionality via an HTML interface.

Monoliths offer some distinct advantages, like:

* Build
* Test
* Deploy
* Troubleshoot
* Scale

How to scale?

#12

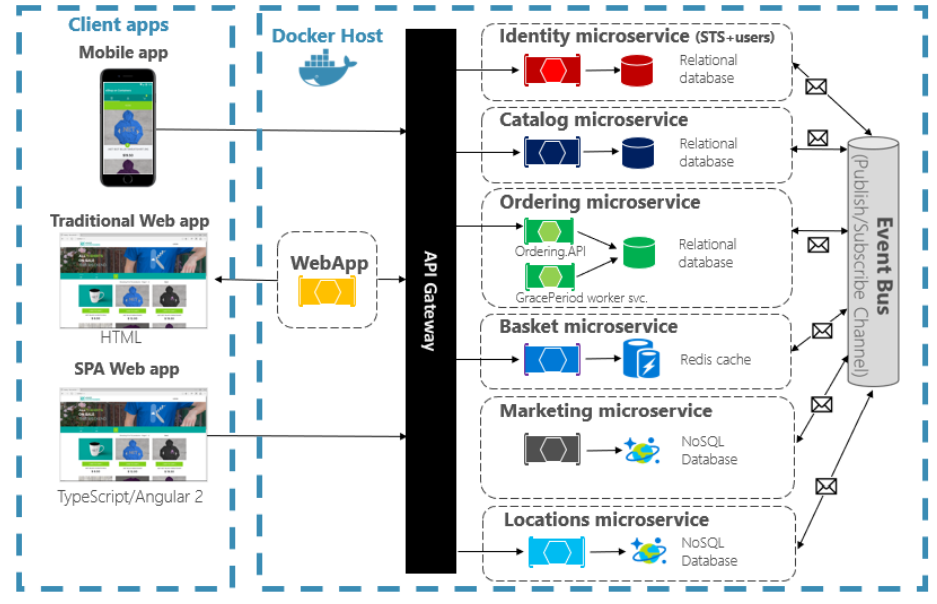
As the app continues to evolve, iteration after iteration, adding more and more functionality.

This develops a Fear Cycle, state in which the developers find themselves losing control.

1. The app has become so overwhelmingly complicated that no single person understands it.
2. Making changes is feared – each change has unintended and costly side effects.
3. New features/fixes become tricky, time-consuming and expensive to implement.
4. Each release as small as possible and requires a full deployment to the entire application.
5. One unstable component can crash the entire system.
6. New technologies and frameworks aren’t an option.
7. It is difficult to implement agile delivery methodologies.
8. Architectural erosion sets in as the code base deteriorates with never-ending “special cases”.
9. The consultant tells you to rewrite it.

This monolithic fear-cycle by adopting a cloud-native approach to building systems.

#13



Features of cloud-native approach:

1. The application is decomposed across a set of small isol