

Visual Studio **LIVE!**  
EXPERT SOLUTIONS FOR .NET DEVELOPERS

San Diego

# W15 Message-Based Microservices Architectures – Driven by Docker

Michele Leroux Bustamante

Microservices / Security Architect

Solliance  
Intermediate



Code Again for  
the First Time!



Visual Studio **LIVE!**  
EXPERT SOLUTIONS FOR .NET DEVELOPERS

San Diego

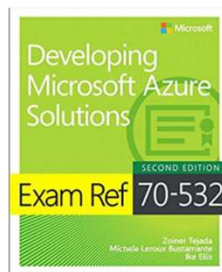
# W15 Message-Based Microservices Architectures – Driven by Docker

Michele Leroux Bustamante  
Microservices / Security Architect

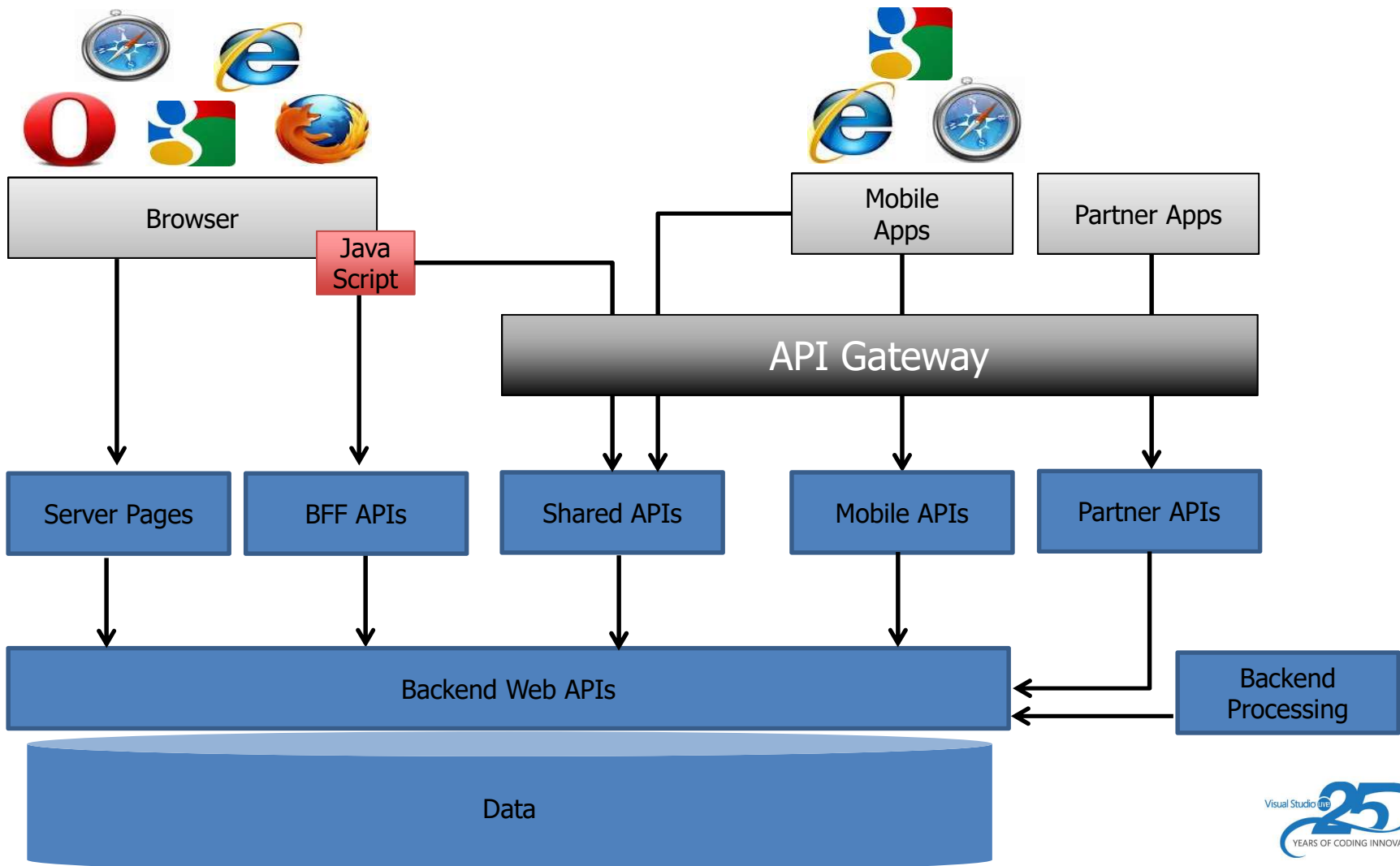
Solliance  
Intermediate



✉ michelebusta@solliance.net  
🐦 @michelebusta

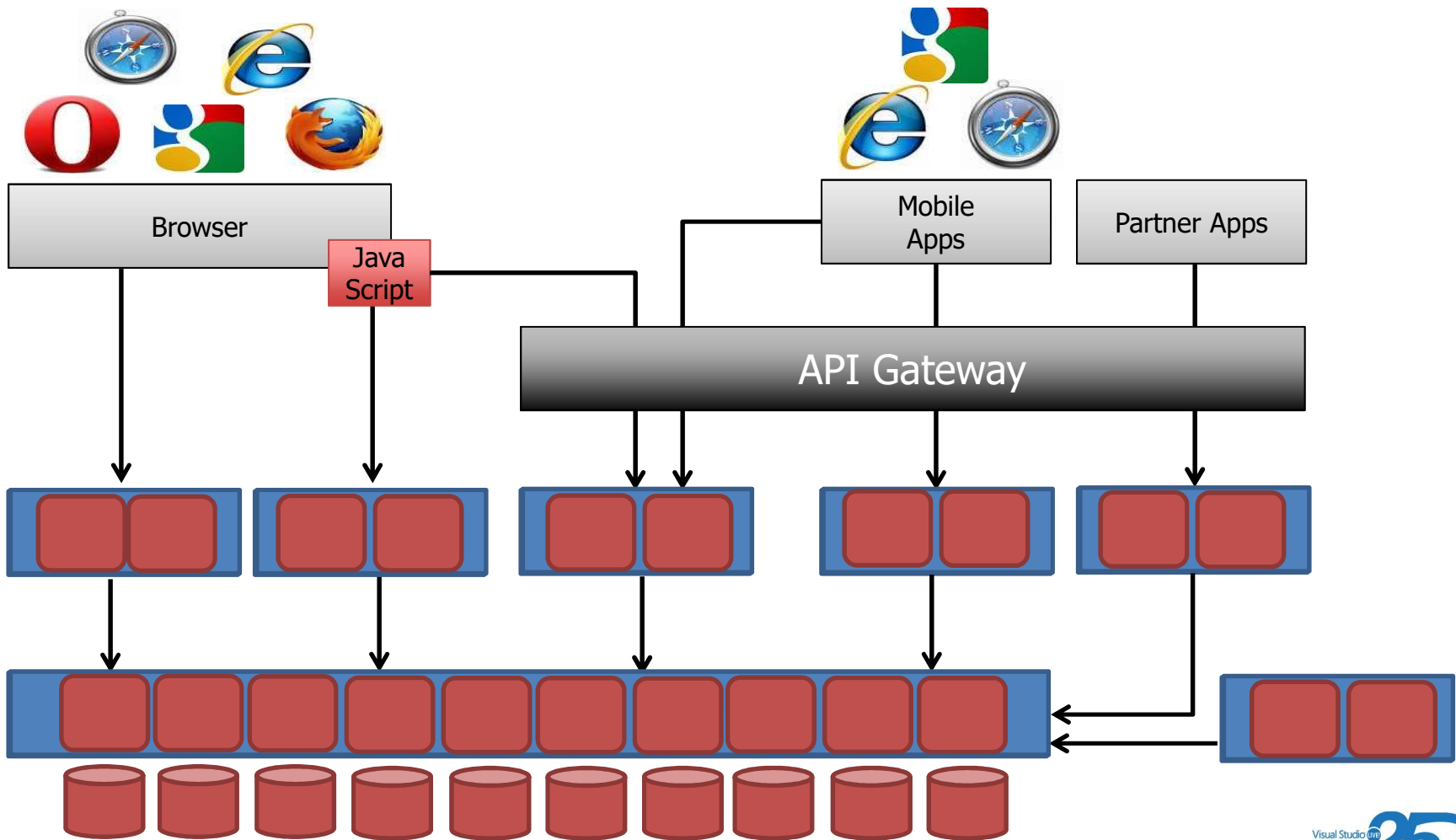


HELLO WORLD  
Today...





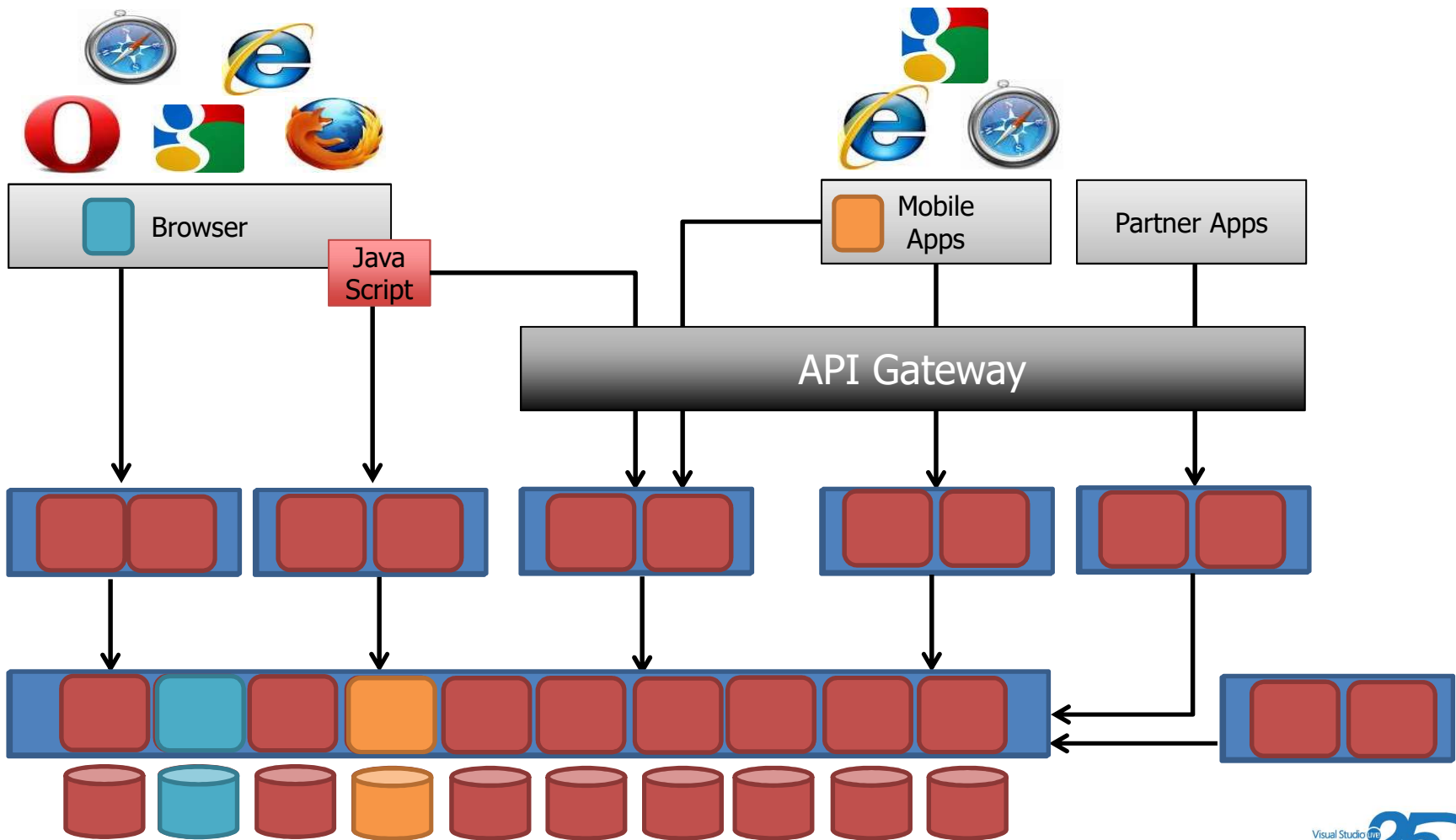
Said.  
No one.  
Ever.







It's all about the business benefits.



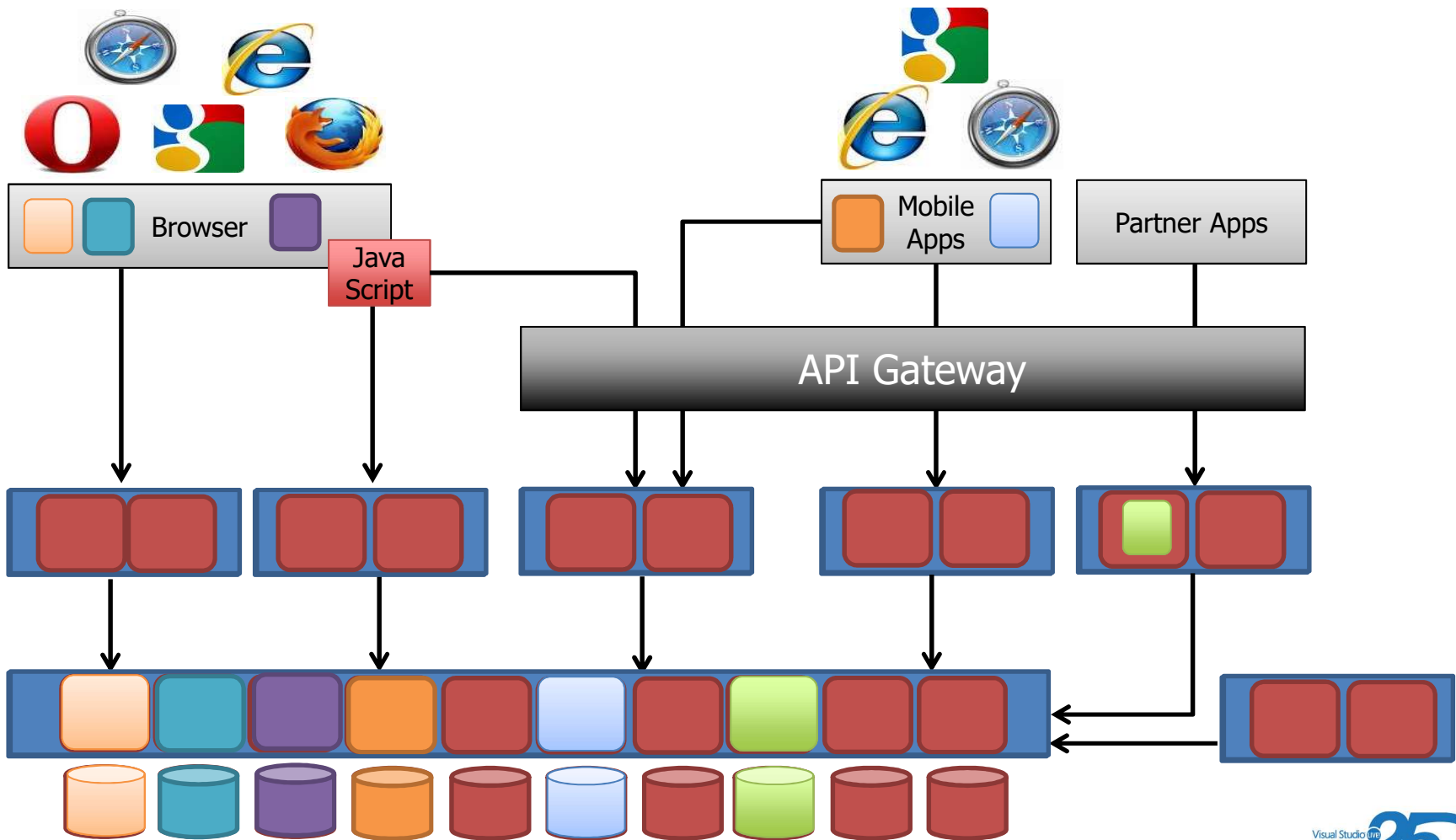


**NEW FEATURES...**



**EVERYWHERE!**

memegenerator.net





Another release?

imgflip.com

JAKE-CLARK.TUMBLR



**SO, TELL ME AGAIN**

**HOW EVERYTHING'S FINE IN  
PRODUCTION!**

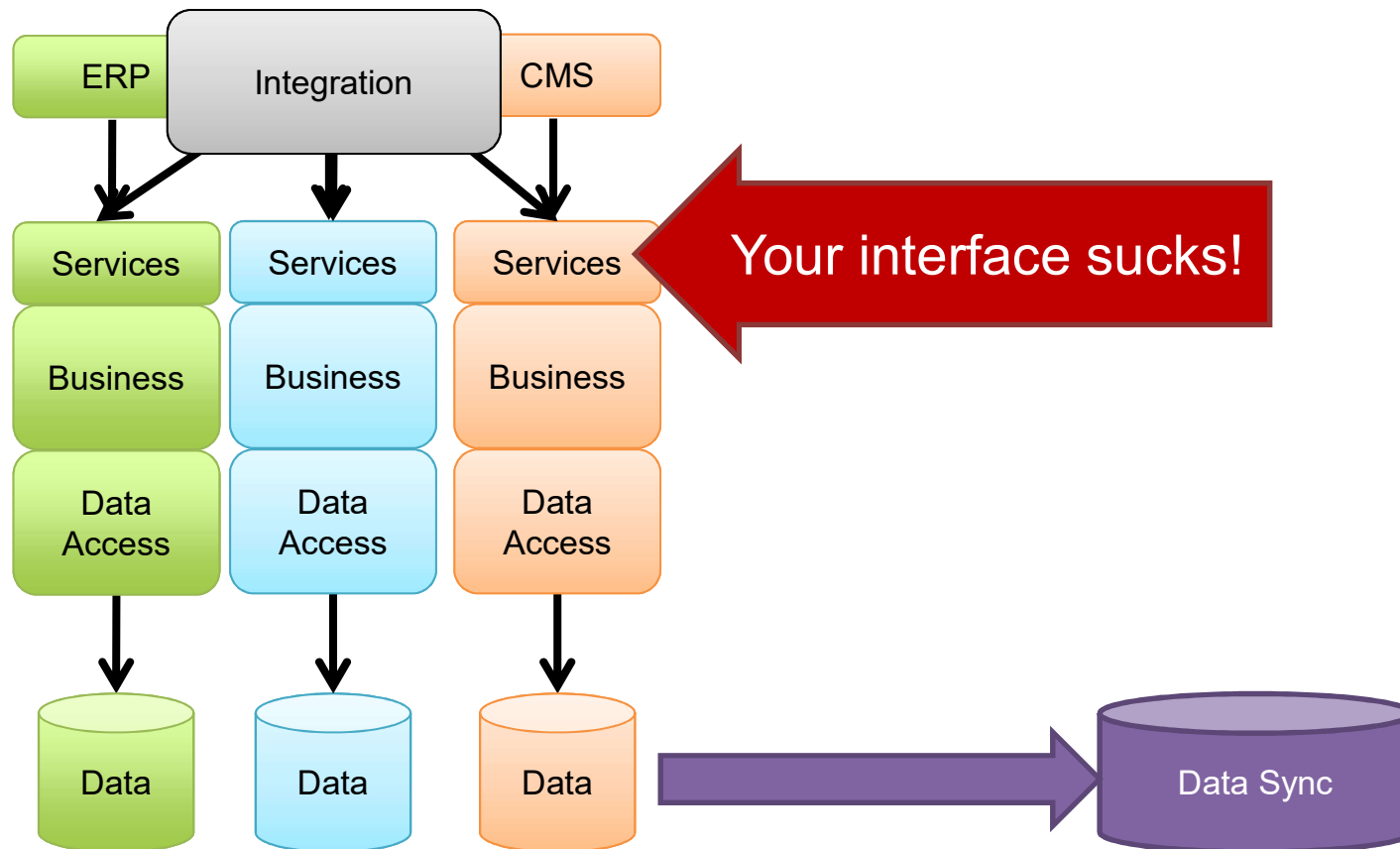
\$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$



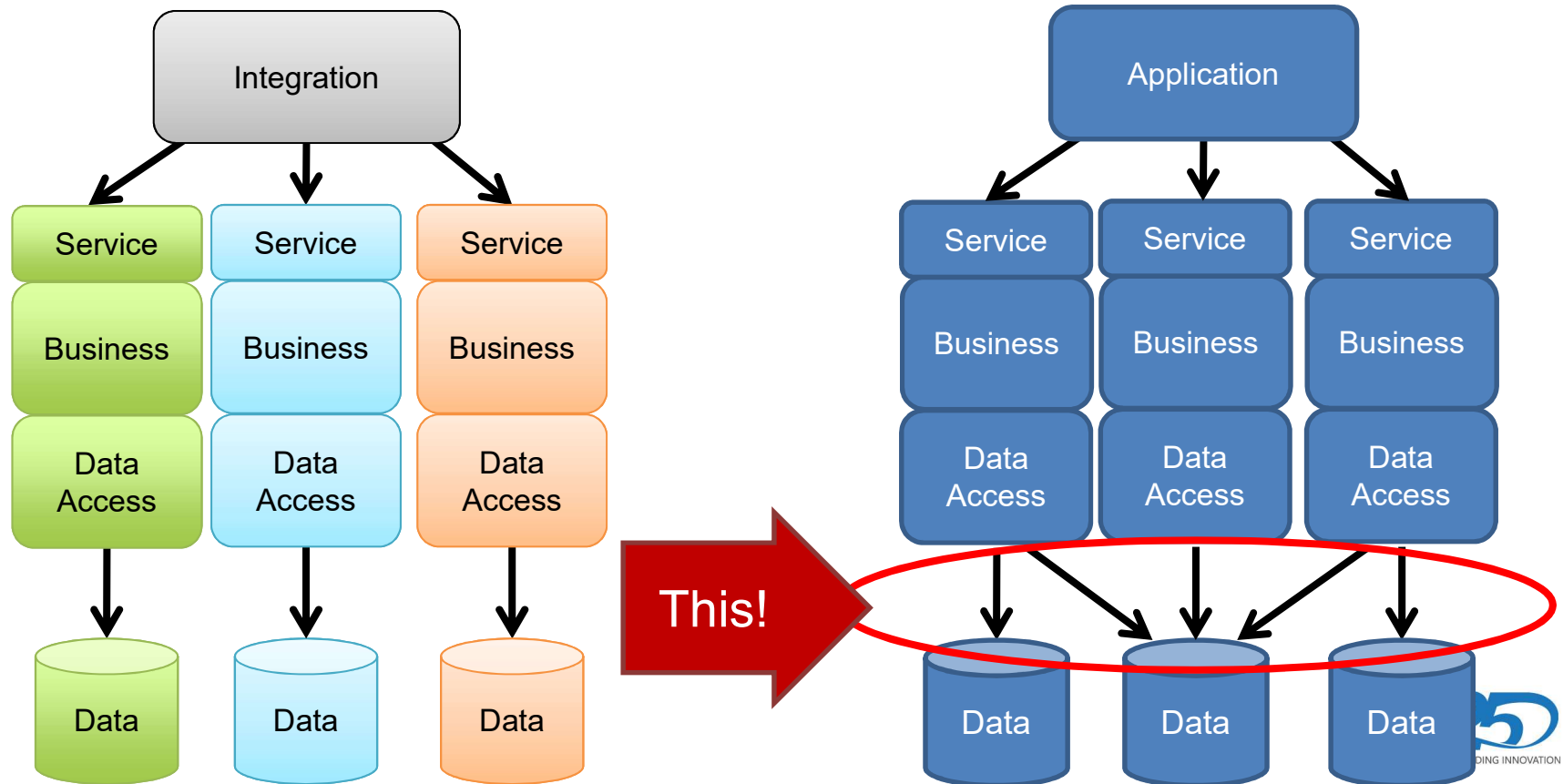
Microservice Architecture  
picks up where SOA left off...



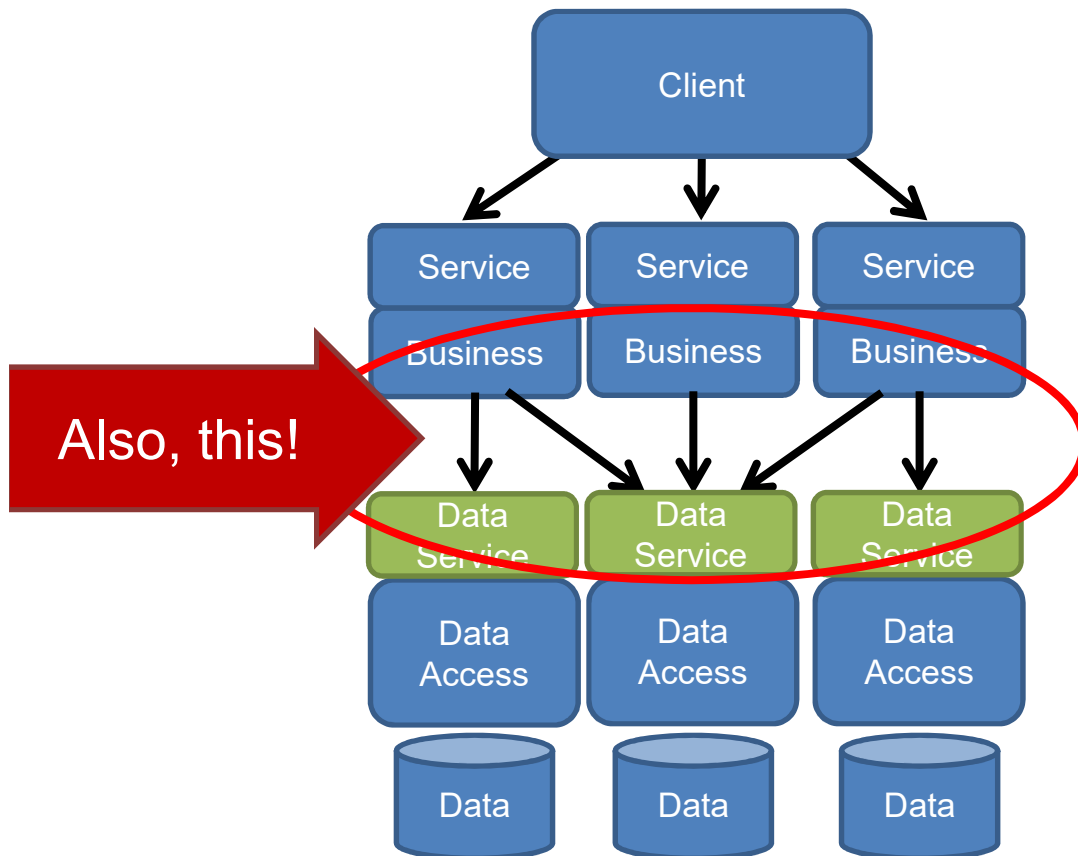
# Big SOA, Little SOA



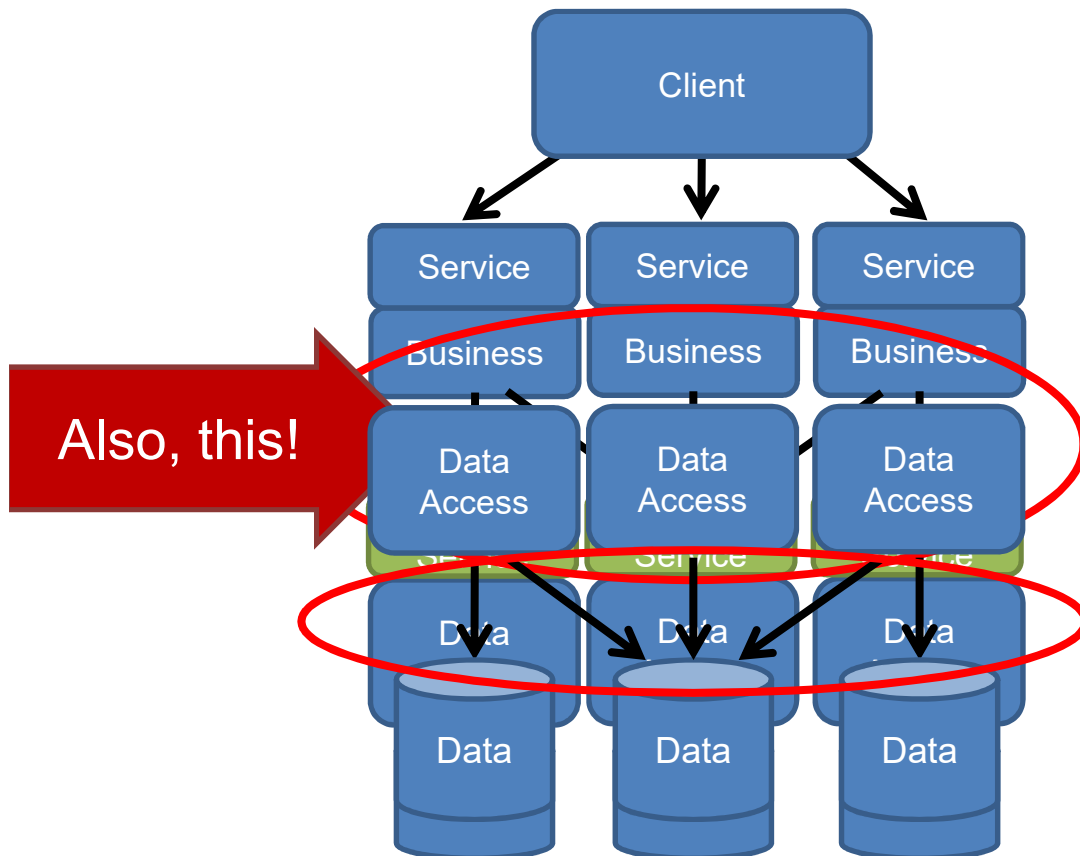
# Big SOA, Little SOA



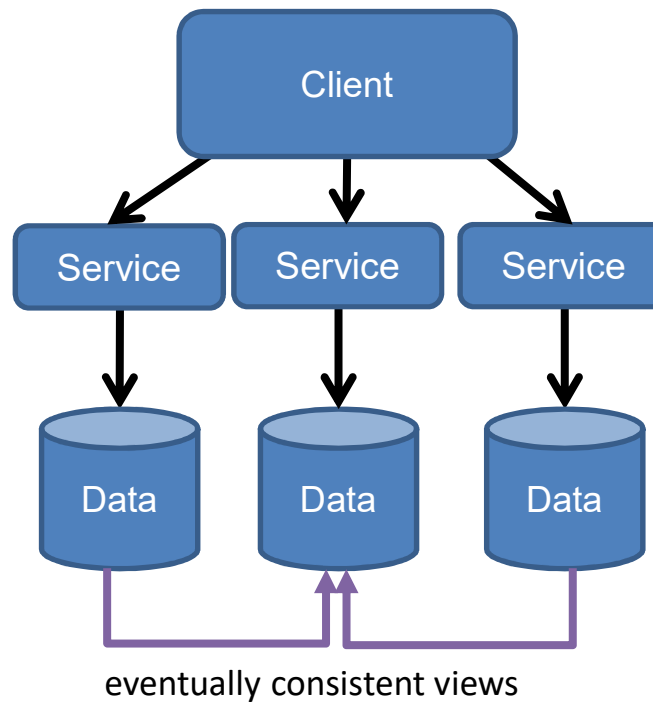
# Solving the Little SOA Issue



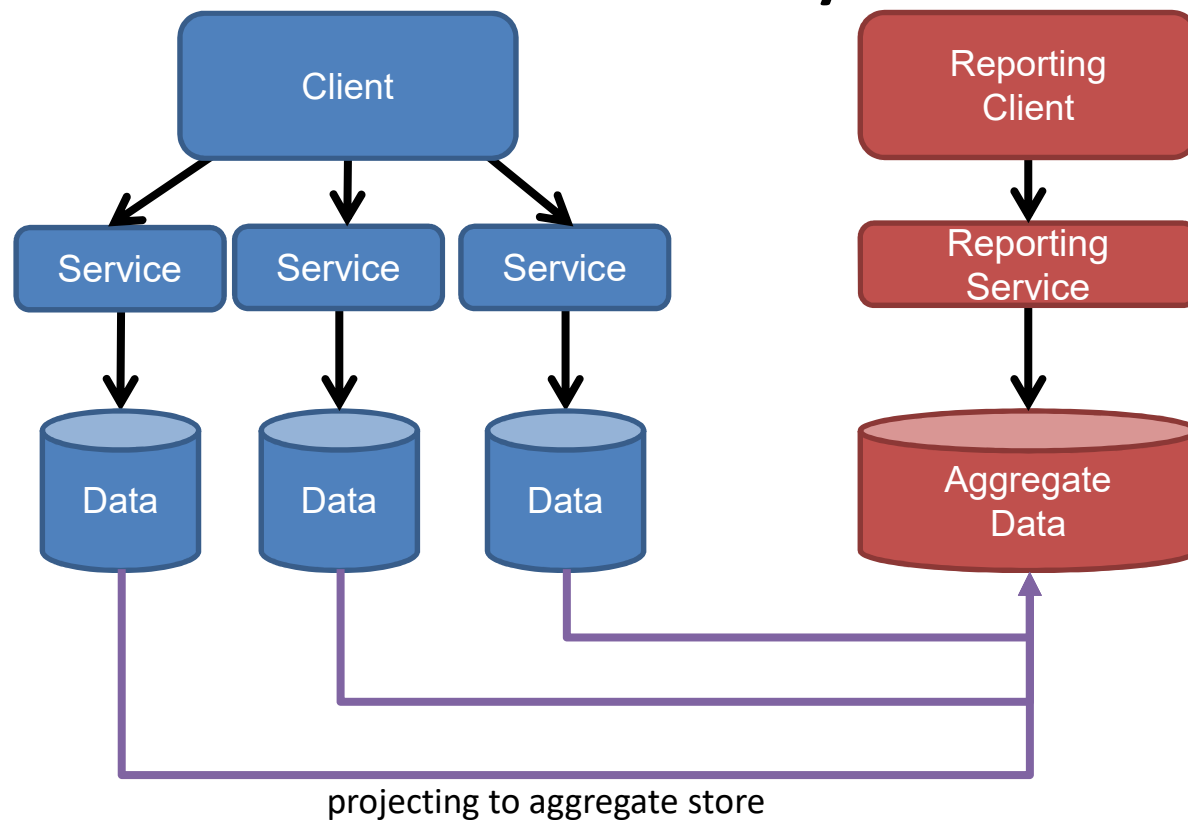
# Solving the Little SOA Issue



# Microservices And Eventual Consistency



# Microservices and Eventual Consistency



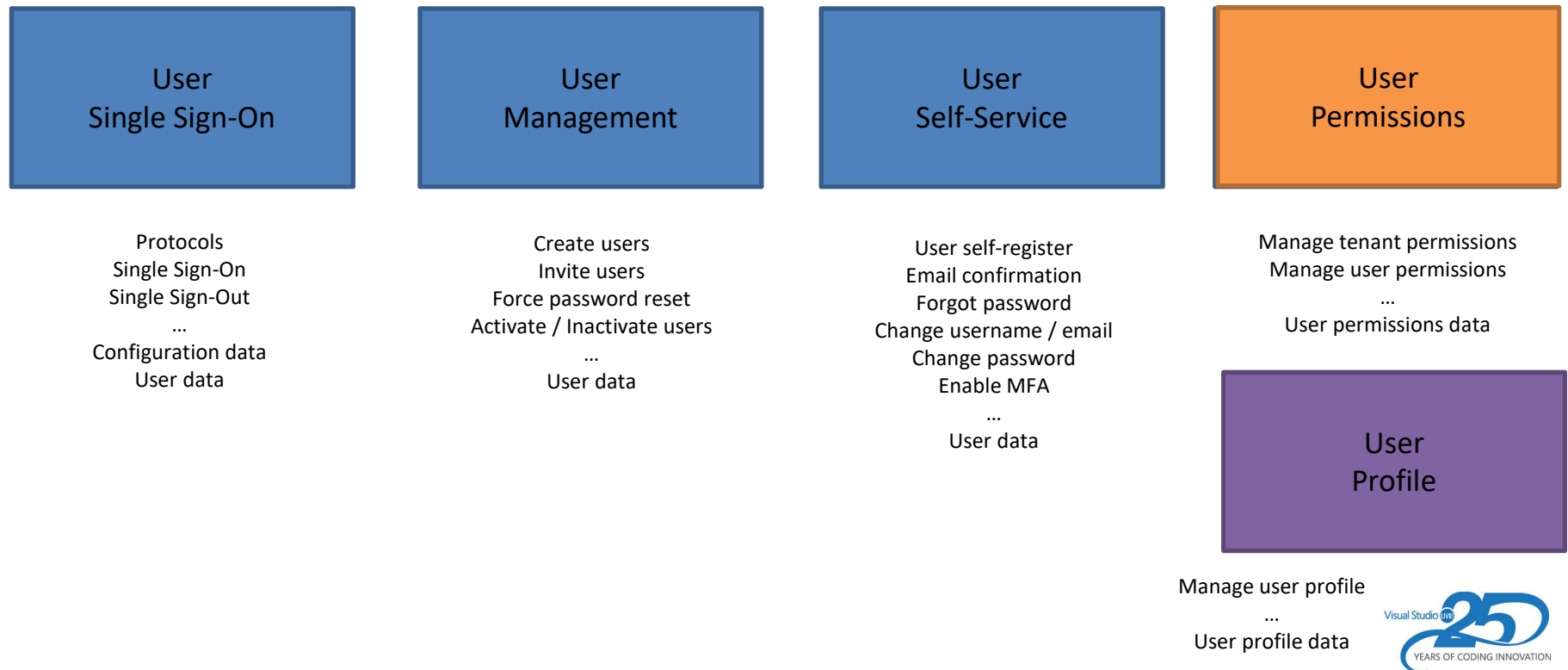


# Design Without Data Constraints

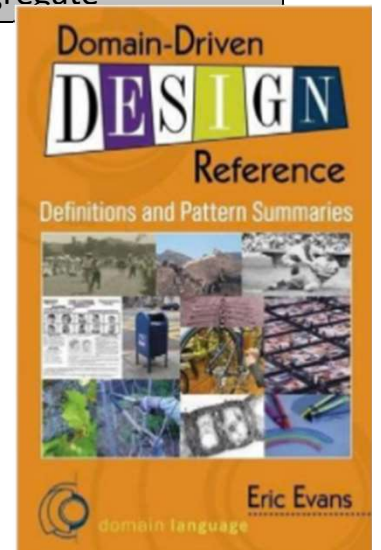
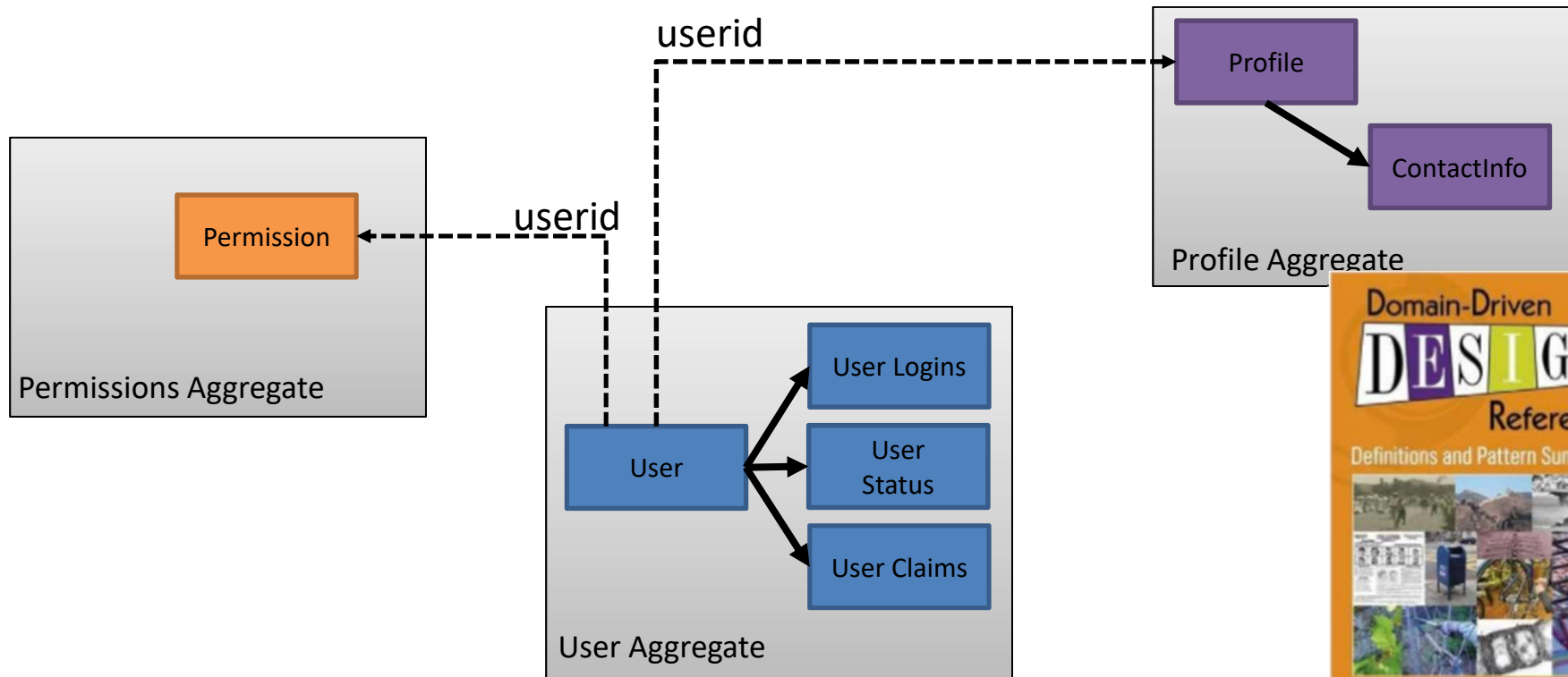
# Identifying Business Domains



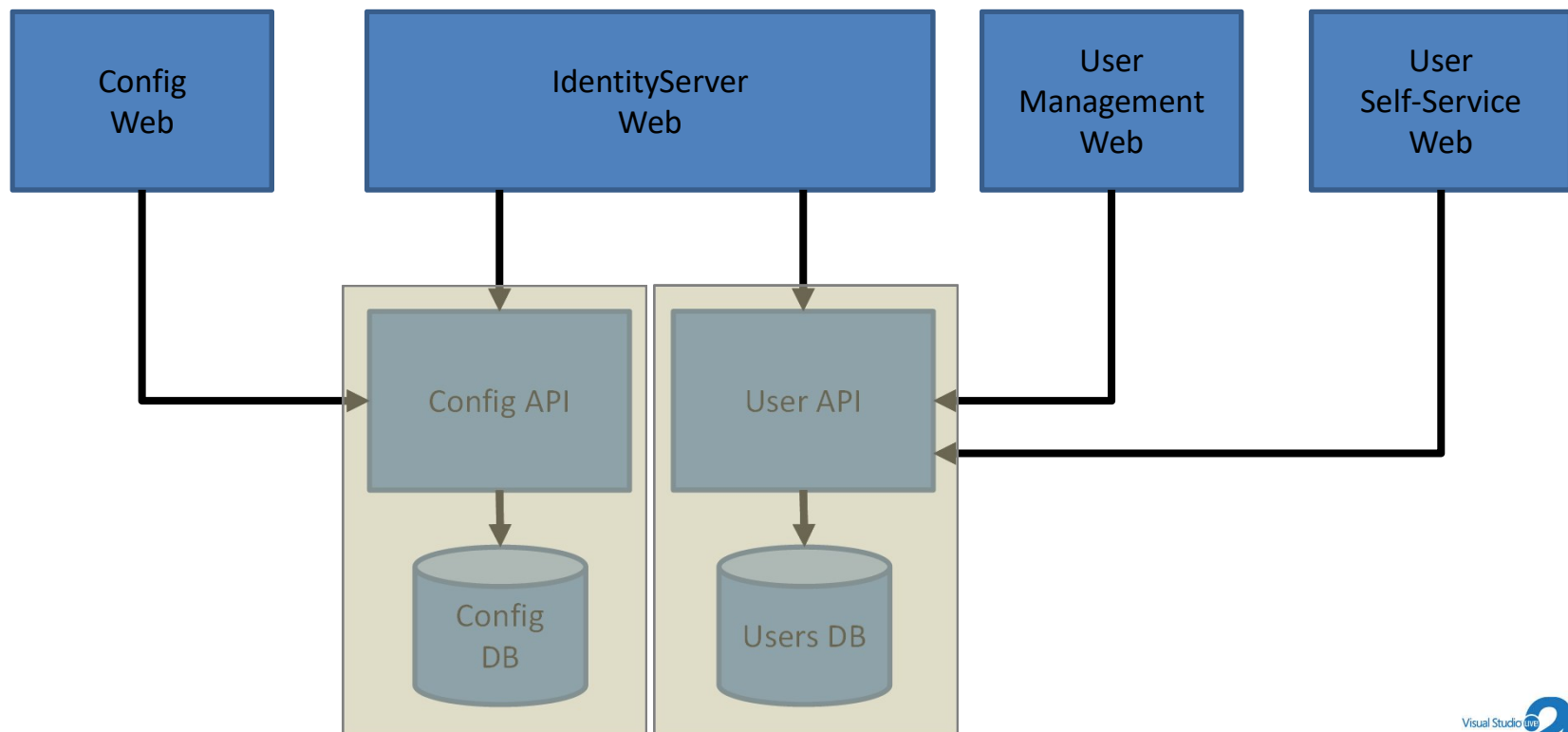
# Identity Management Domain



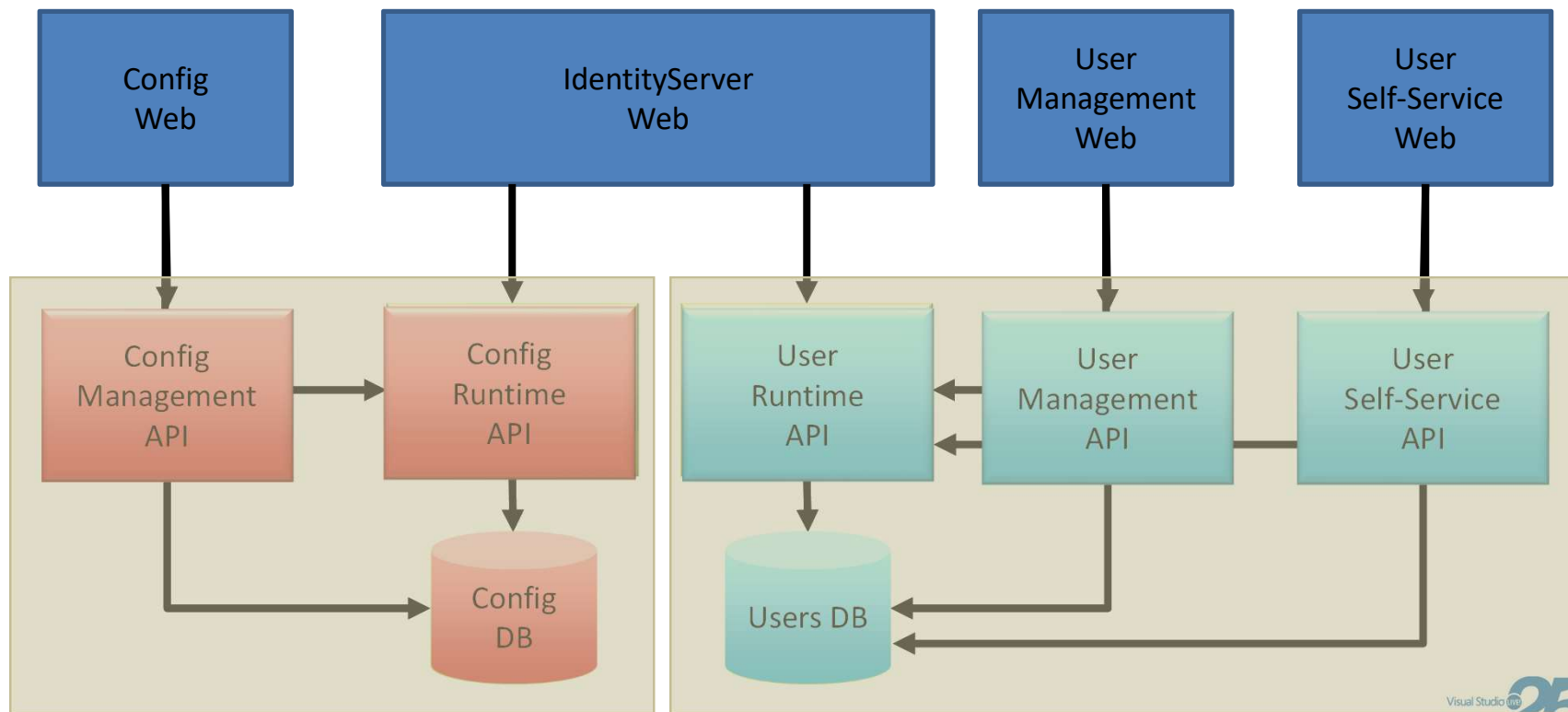
# Aggregates



# Shared APIs

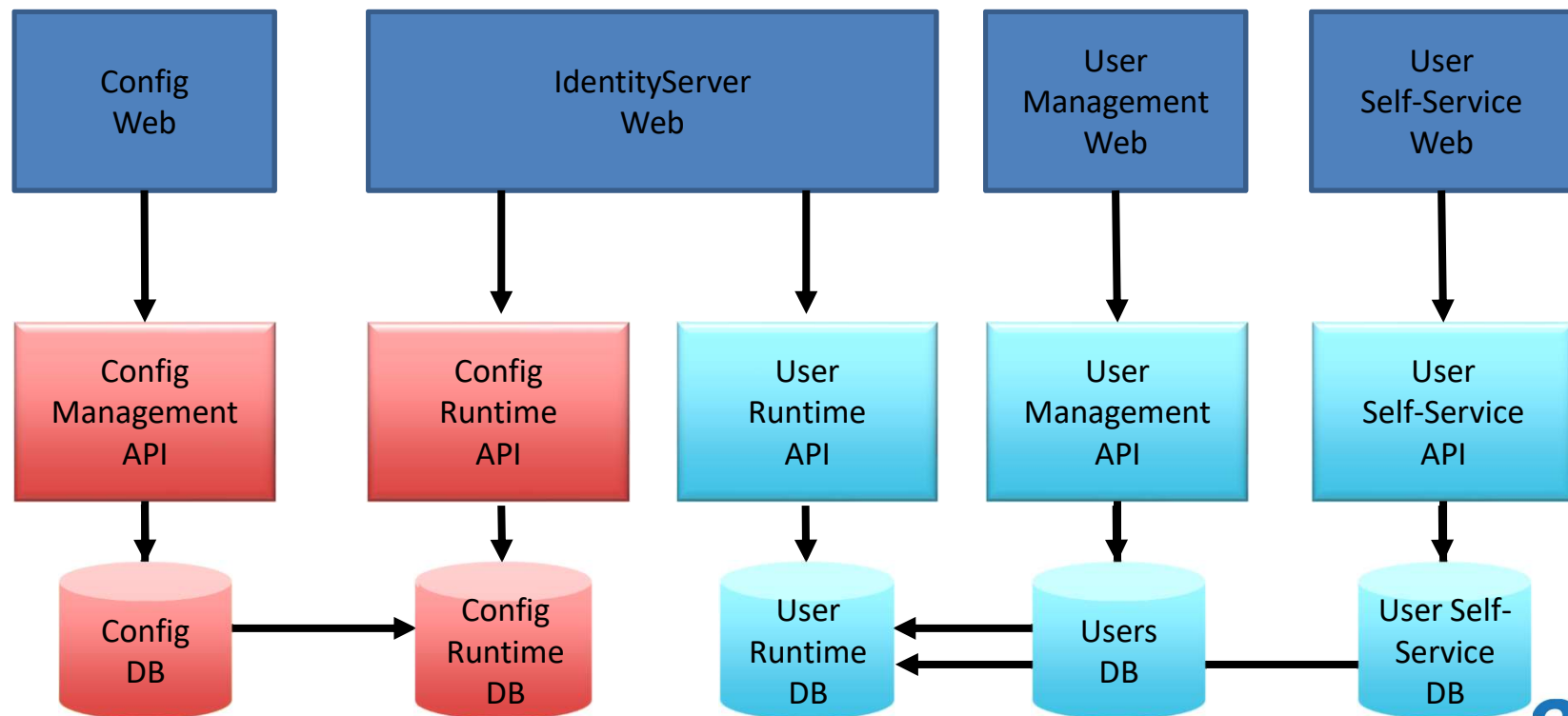


# Shared Aggregates

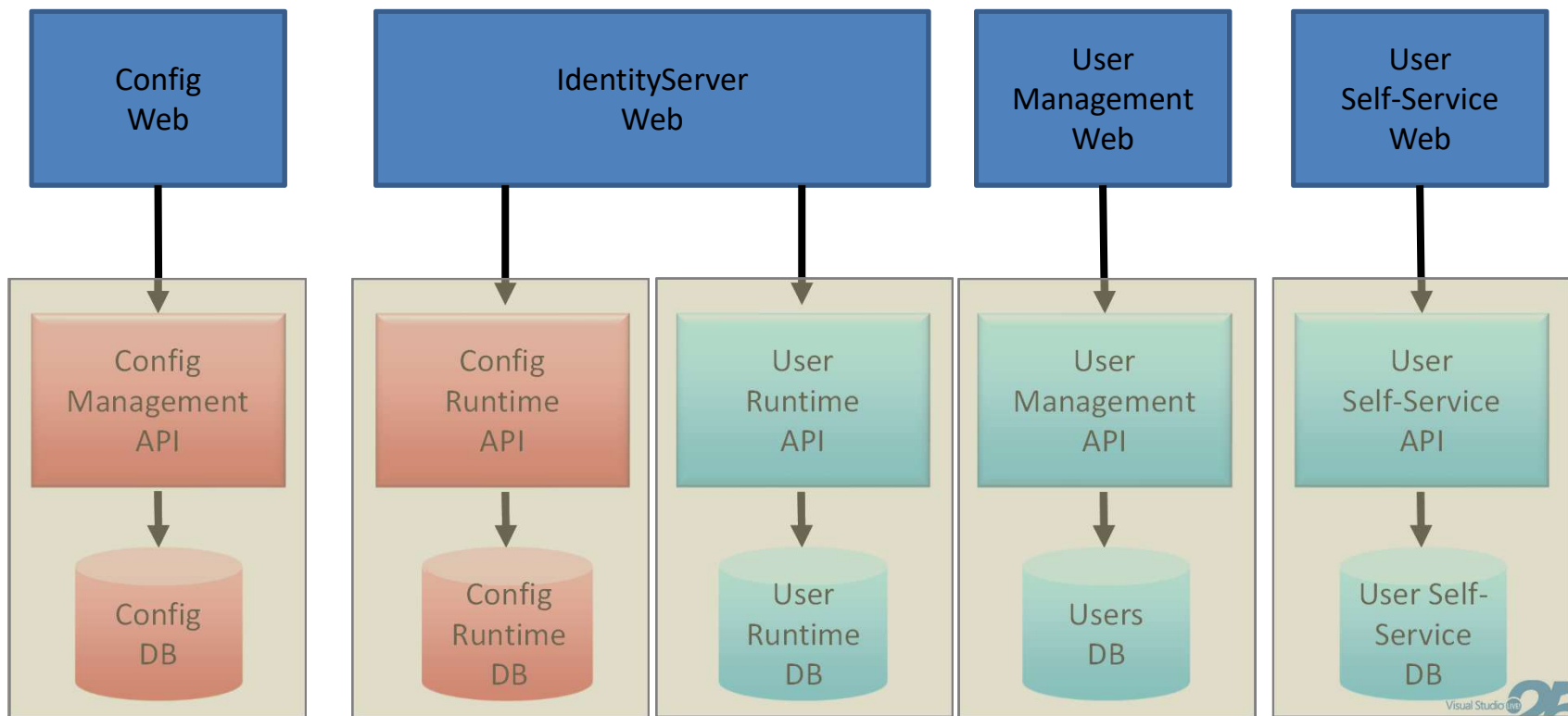




# Full stack isolation

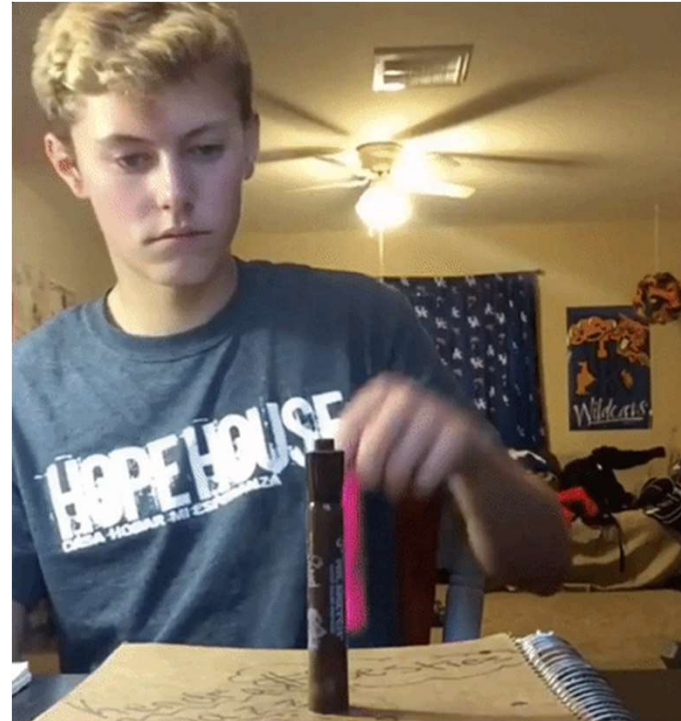


# Full stack isolation

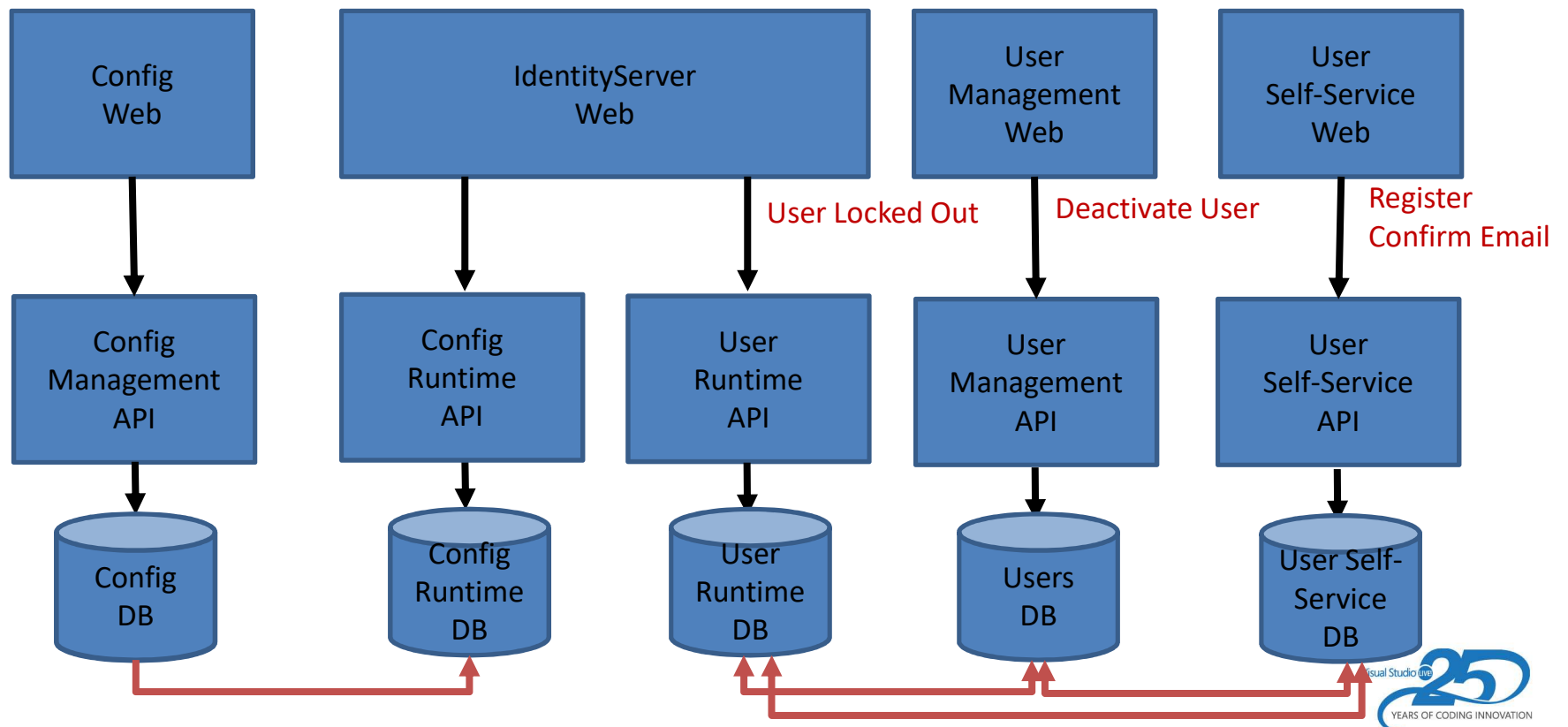


Achievement  
unlocked...

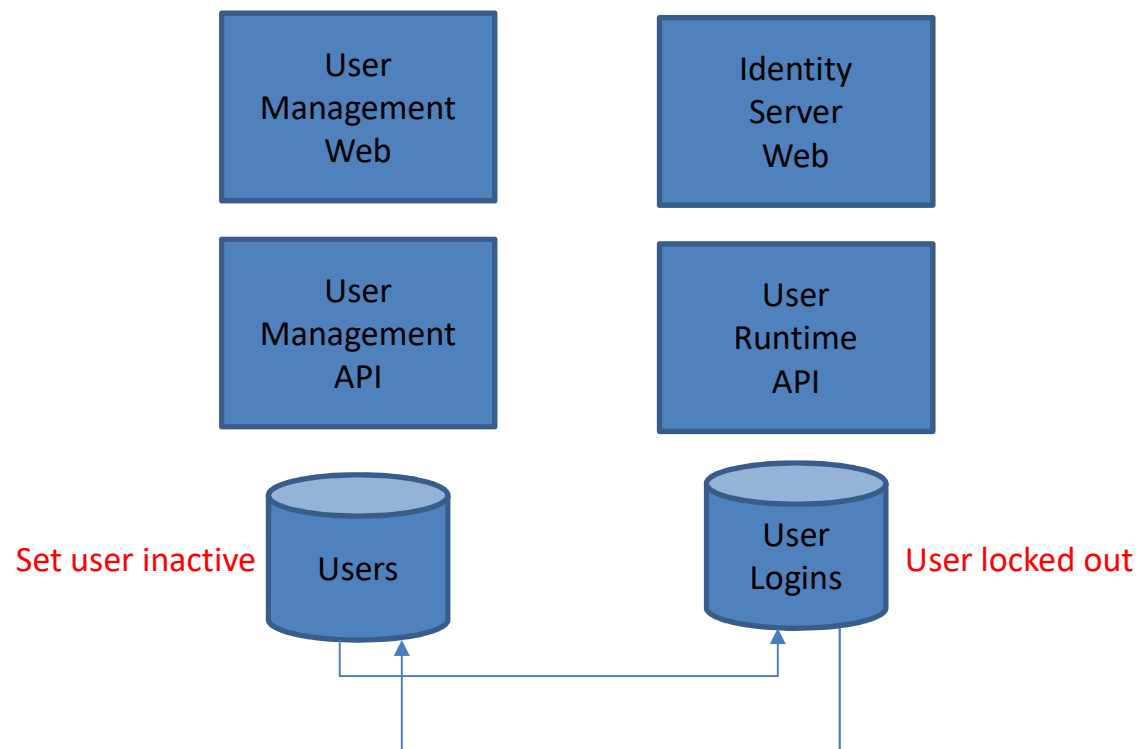
every service  
owns its data



# Eventual consistency



# Row Change Projections



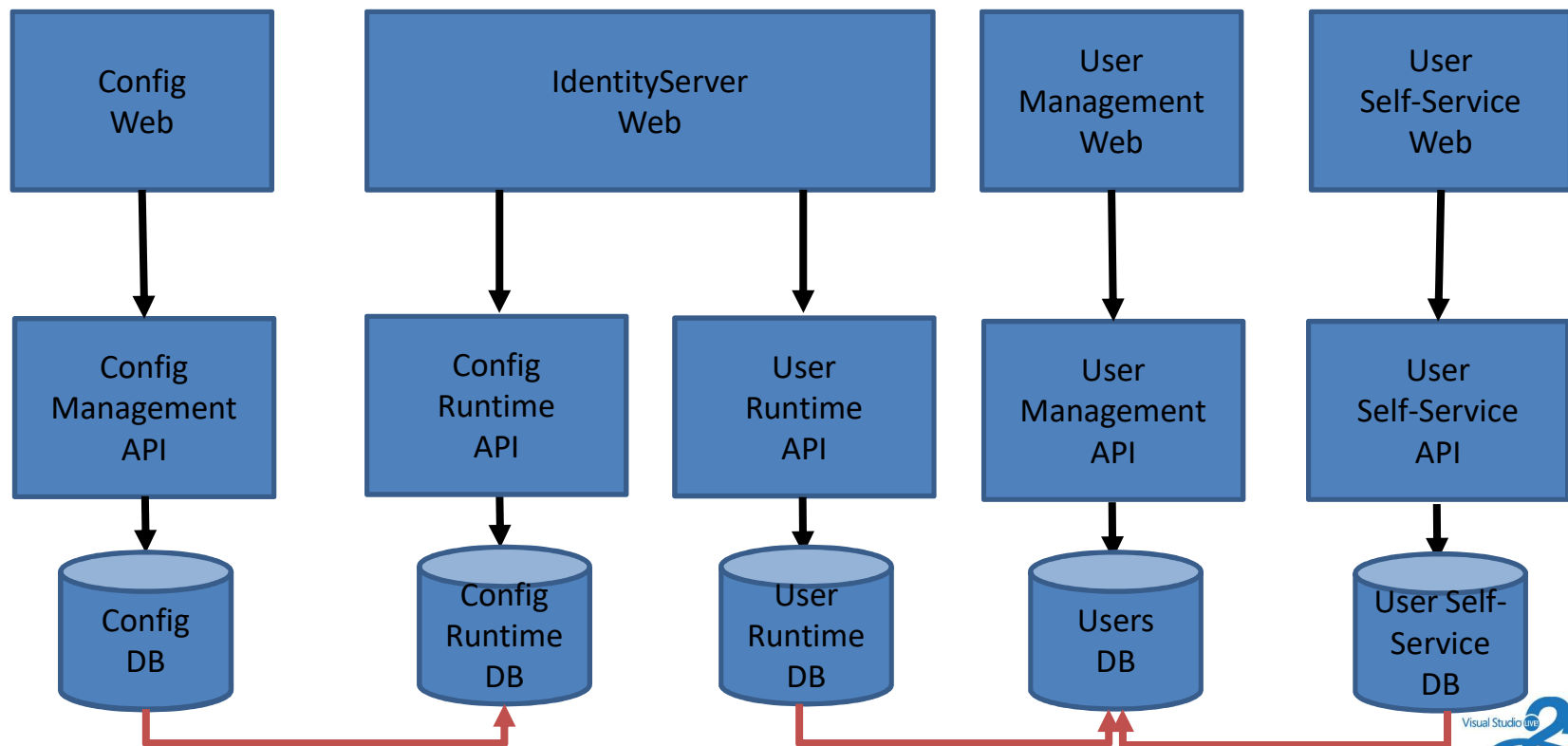
How to create new projections  
later?

How to preserve history?

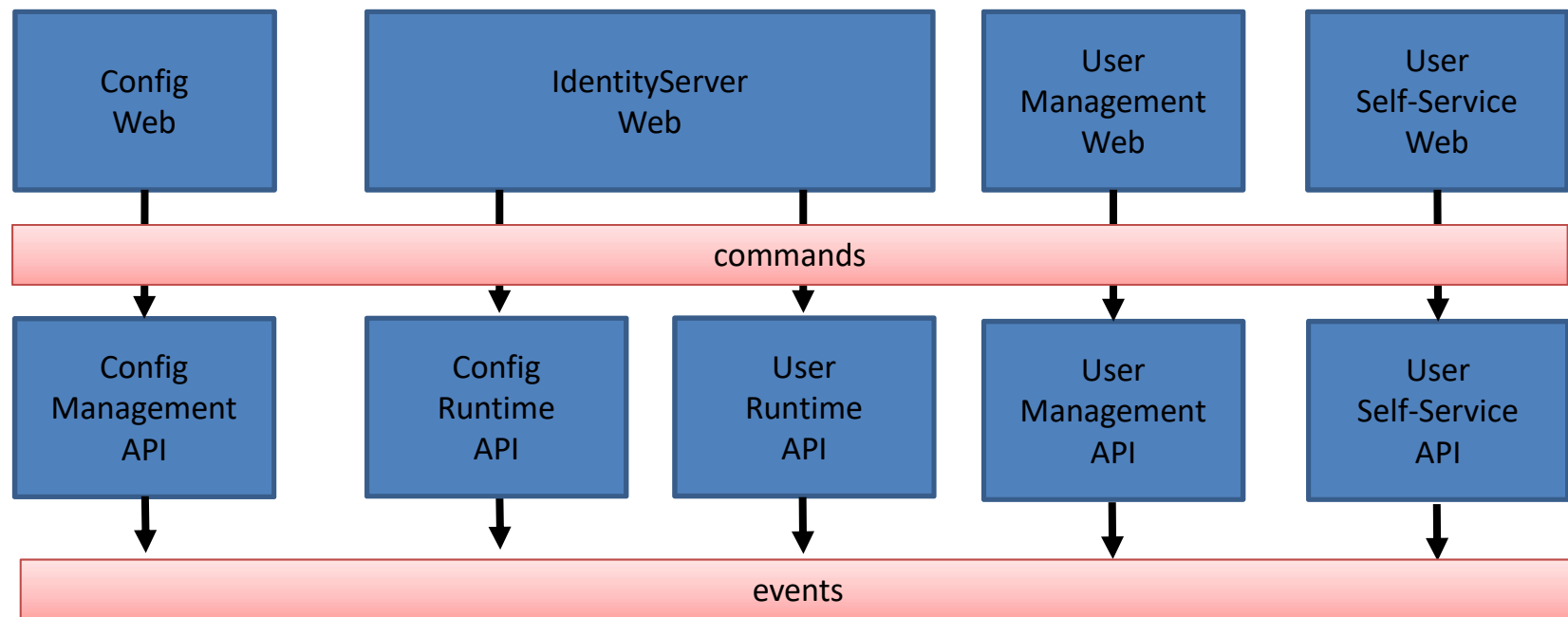
How to retain audits?



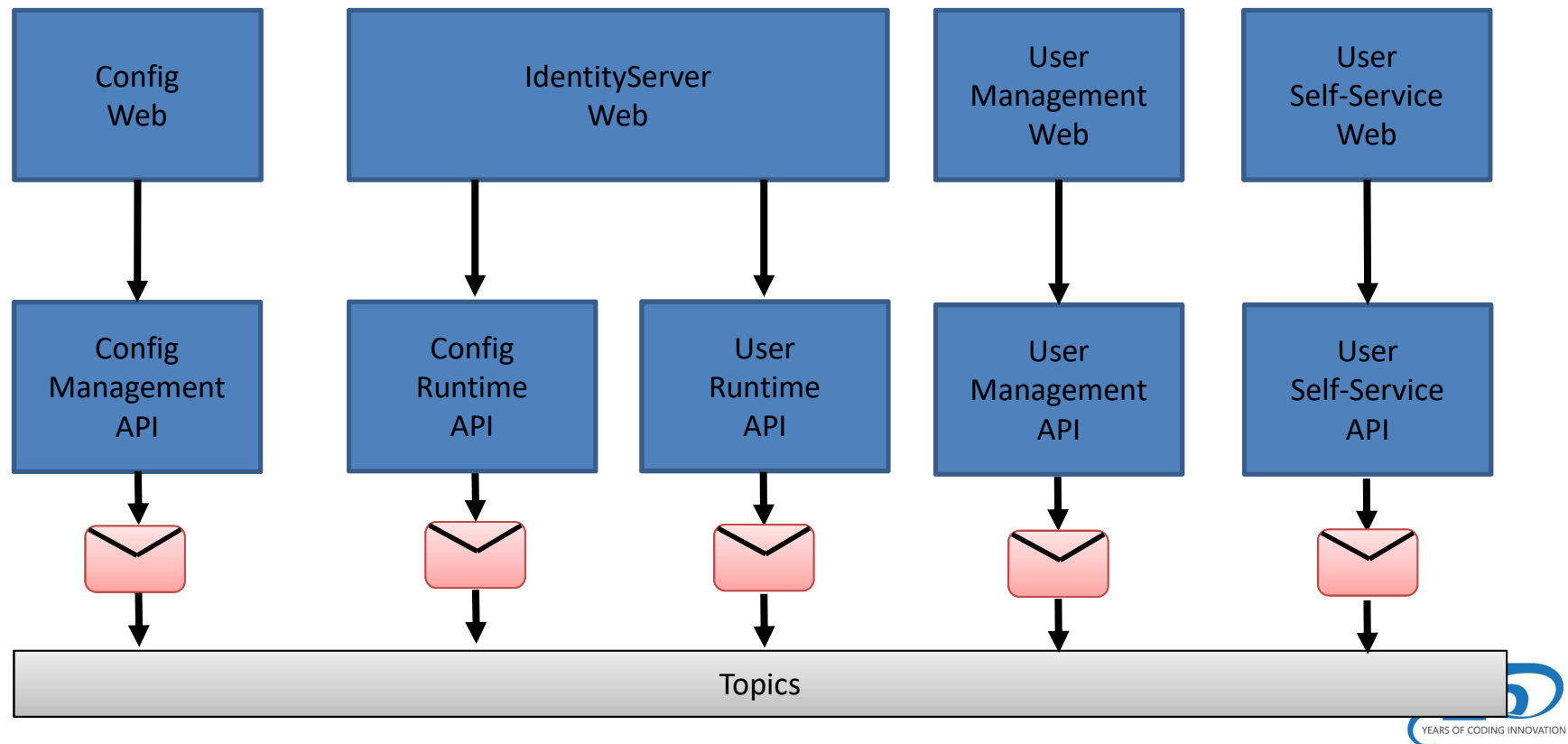
# Messaging-based solution



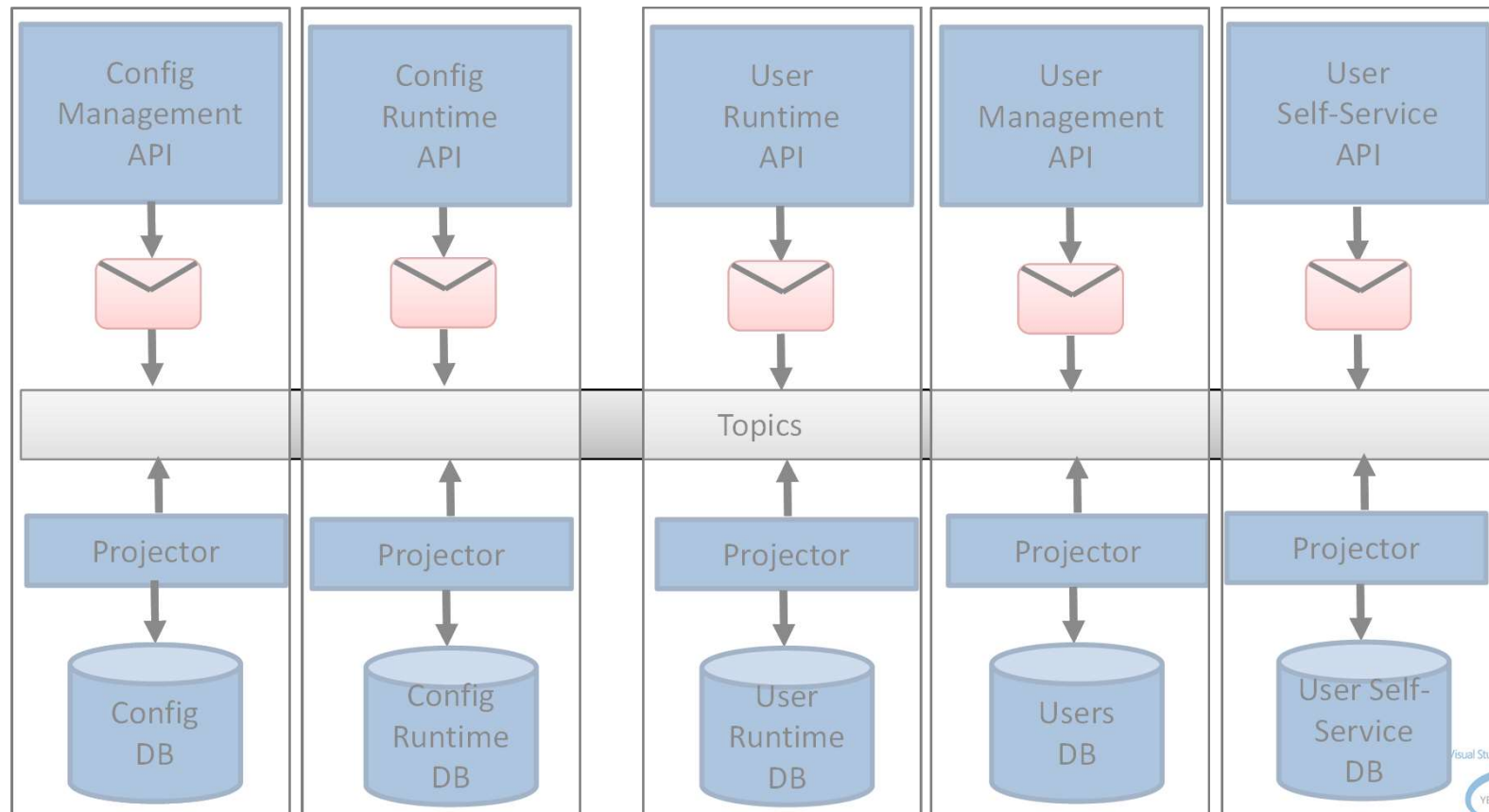
# Messaging-based solution



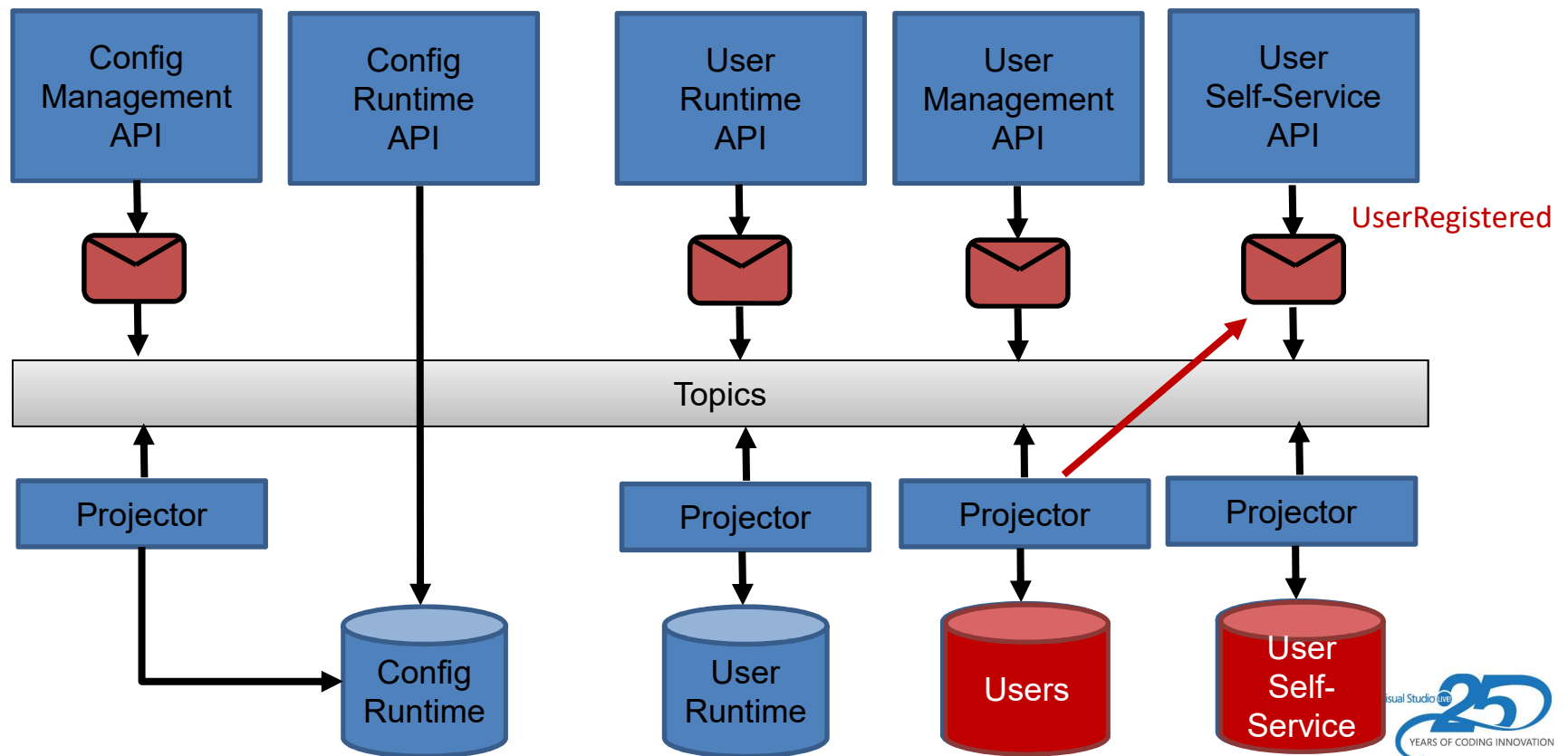
# Messaging-based solution



# Projecting data



# CQRS Views



Messaging Strategy  
Local and Cloud Development

**DEMO**

If you don't write logs  
you won't have them...

-Captain Obvious

Necessary?

February 5th 2018, 09:12:19.643	Executed DbCommand ("4"ms) [Parameters=["@p16='?' (Size = 450), @p0='?' (Size = 4000), @p1='?' (Size = 4000), @p17='?' (Size = 4000), @p2='?' (Size = 256), @p3='?' (Size = 256), @p4='?' (Size = 4000), @p5='?' (Size = 4000), @p6='?' (Size = 4000), @p7='?' (Size = 256), @p8='?' (Size = 256), @p9='?' (Size = 256), @p10='?' (Size = 4000), @p11='?' (Size = 4000), @p12='?' (Size = 4000), @p13='?' (Size = 4000), @p14='?' (Size = 256), @p15='?' (Size = 256)"], CommandType='Text', CommandTimeout='30'] ""SET NOCOUNT ON; UPDATE [appidentity].[AspNetUsers] SET [AccessFailedCount] = @p0, [ConcurrencyStamp] = @p1, [Email] =
February 5th 2018, 09:12:19.635	Executed DbCommand ("8"ms) [Parameters=["@__normalizedUserName_0='?' (Size = 256)"], CommandType='Text', CommandTimeout='30'] ""SELECT TOP(1) [u].[Id], [u].[AccessFailedCount], [u].[ConcurrencyStamp], [u].[Email], [u].[EmailConfirmed], [u].[Firstname], [u].[Lastname], [u].[LockoutEnabled], [u].[LockoutEnd], [u].[NormalizedEmail], [u].[NormalizedUserName], [u].[PasswordHash], [u].[PhoneNumber], [u].[PhoneNumberConfirmed], [u].[SecurityStamp], [u].[TwoFactorEnabled], [u].[UserName] FROM [appidentity].[AspNetUsers] AS [u]
February 5th 2018, 09:12:19.611	Executed DbCommand ("6"ms) [Parameters=["@__request_SubjectId_0='?' (Size = 450)"], CommandType='Text', CommandTimeout='30'] ""SELECT TOP(1) [uu].[Id], [uu].[AccessFailedCount], [uu].[ConcurrencyStamp], [uu].[Email], [uu].[EmailConfirmed], [uu].[Firstname], [uu].[Lastname], [uu].[LockoutEnabled], [uu].[LockoutEnd], [uu].[NormalizedEmail], [uu].[NormalizedUserName], [uu].[PasswordHash], [uu].[PhoneNumber], [uu].[PhoneNumberConfirmed], [uu].[SecurityStamp], [uu].[TwoFactorEnabled], [uu].[UserName] FROM [appidentity].[AspNetUsers] AS [uu]
February 5th 2018, 09:12:19.590	Entity Framework Core "2.0.0-rtm-26452" initialized ""AppIdentityDbContext"" using provider ""Microsoft.EntityFrameworkCore.SqlServer"" with options: ""MigrationsAssembly=IdentityUser.Core ""
February 5th 2018, 09:12:19.589	EmailVerifyRequest: { "subjectId": "ada2b33b1e414722a65c541838e47c06", "Password": "BetaPassword!", "Token": "CfdJ8NEXk6dbqZpFmZjT+8Ei9tclAb/h2+1CEzx+C+t1P4qn7OGp72KY8oYhK2ZGbcS54I1YkN92bScjP13Nbhv7Kns0vFD8z+RckxfiFH2/U9odCEEXTQLHuwGXY66Kv8t67TBwimGwp7oCKUIaLmDxRoiVDrE/OPDO6G0JLspT0LYH6omsN4bgWTL4lEMhsUebdH+okH9OWGIut81guVwvVOMaEOvHUHA4e4pZpXVNo47G38C8WhAr1vXuUDhwIXPZKW==" }
February 5th 2018, 09:12:19.585	EmailVerifyAsync
February 5th 2018, 09:12:19.566	Executing action method "Identity.User.Profile.Controllers.SignupController.EmailVerify (Identity.User.Profile)" with arguments ([ "Identity.User.Core.Models.EmailVerifyRequest" ]) - ModelState is valid
February 5th 2018, 09:12:19.506	Starting http request
February 5th 2018, 09:12:19.502	Calling email verify api
February 5th 2018, 09:12:19.491	Executing action method "Identity.User.Profile.Web.Controllers.AccountController.EmailConfirm (Identity.User.Profile.web)" with arguments ([ "Identity.User.Profile.Web.Models.ServiceModels.EmailVerifyRequest" ]) - ModelState is valid

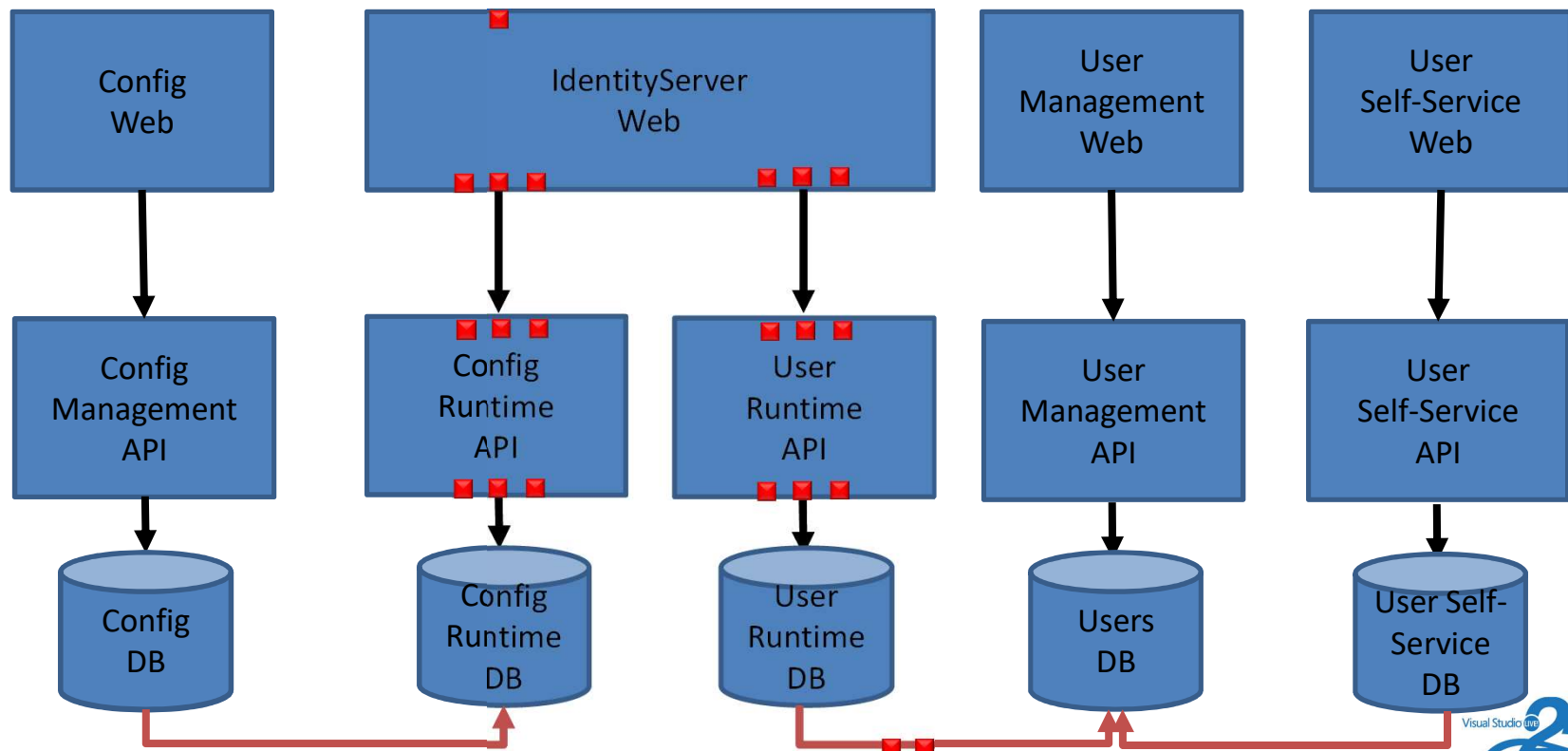
Oh dear

Poor man's solution to end to end

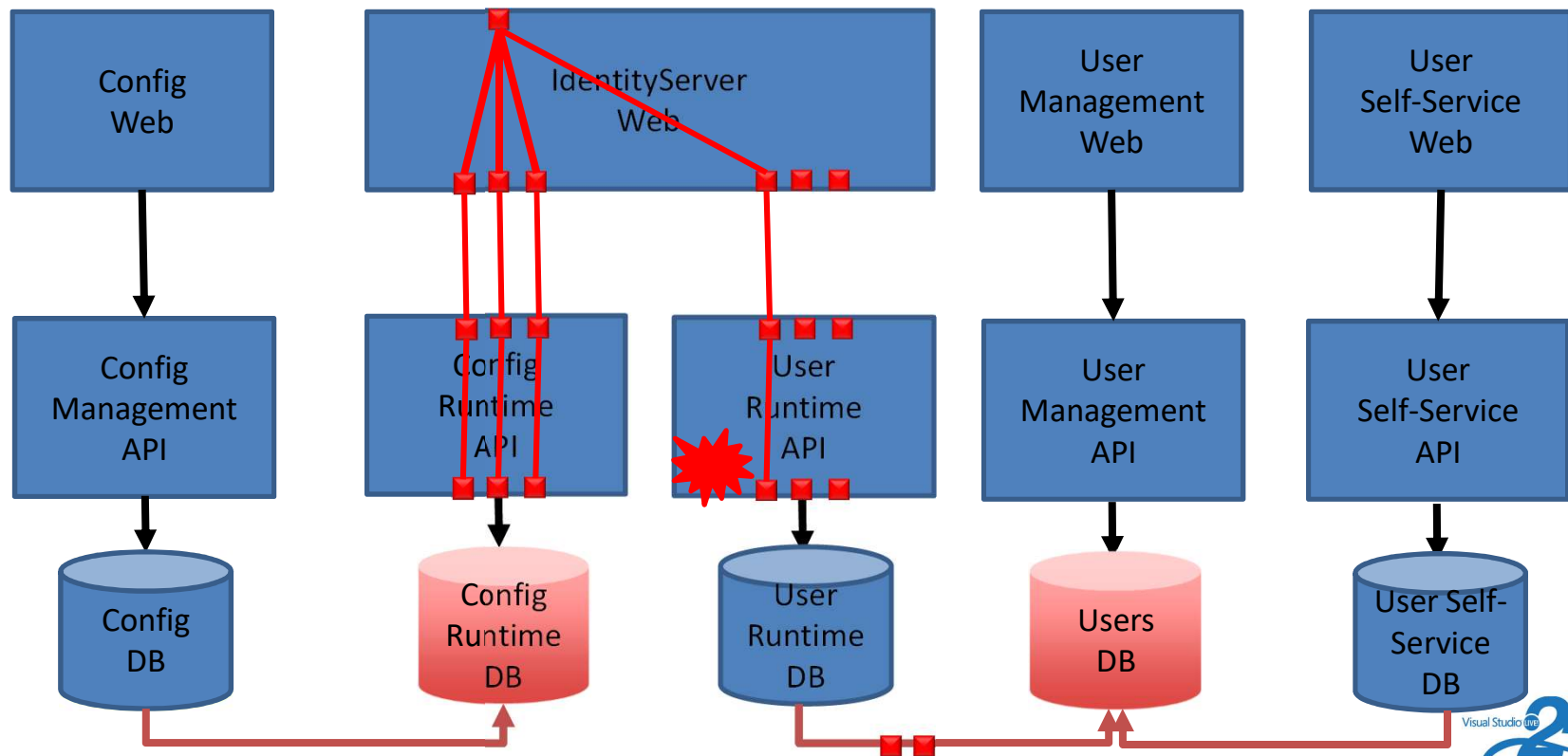
Start of activity



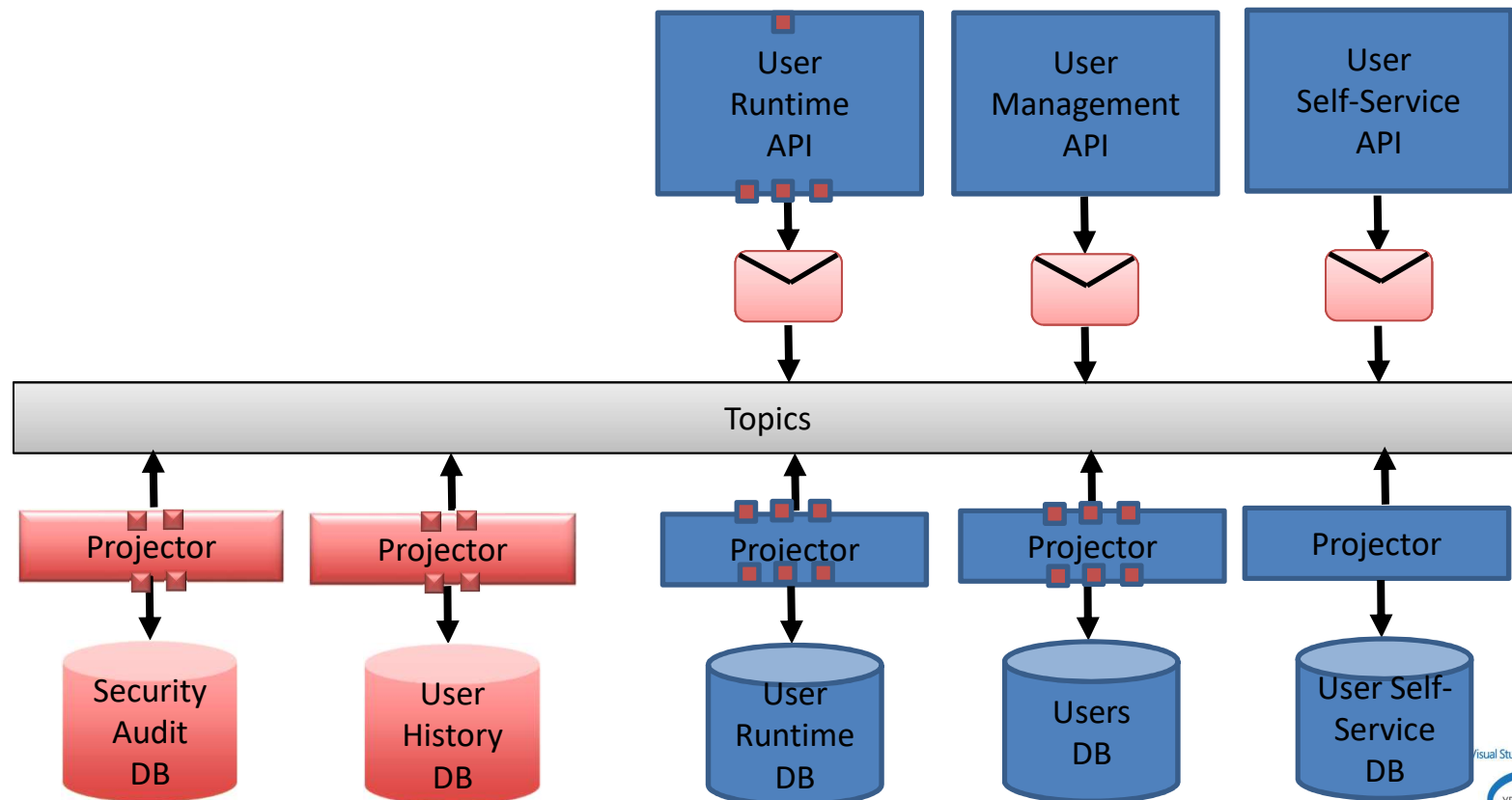
# End to end tracing



# End to end visibility



# Messaging and end-to-end visibility



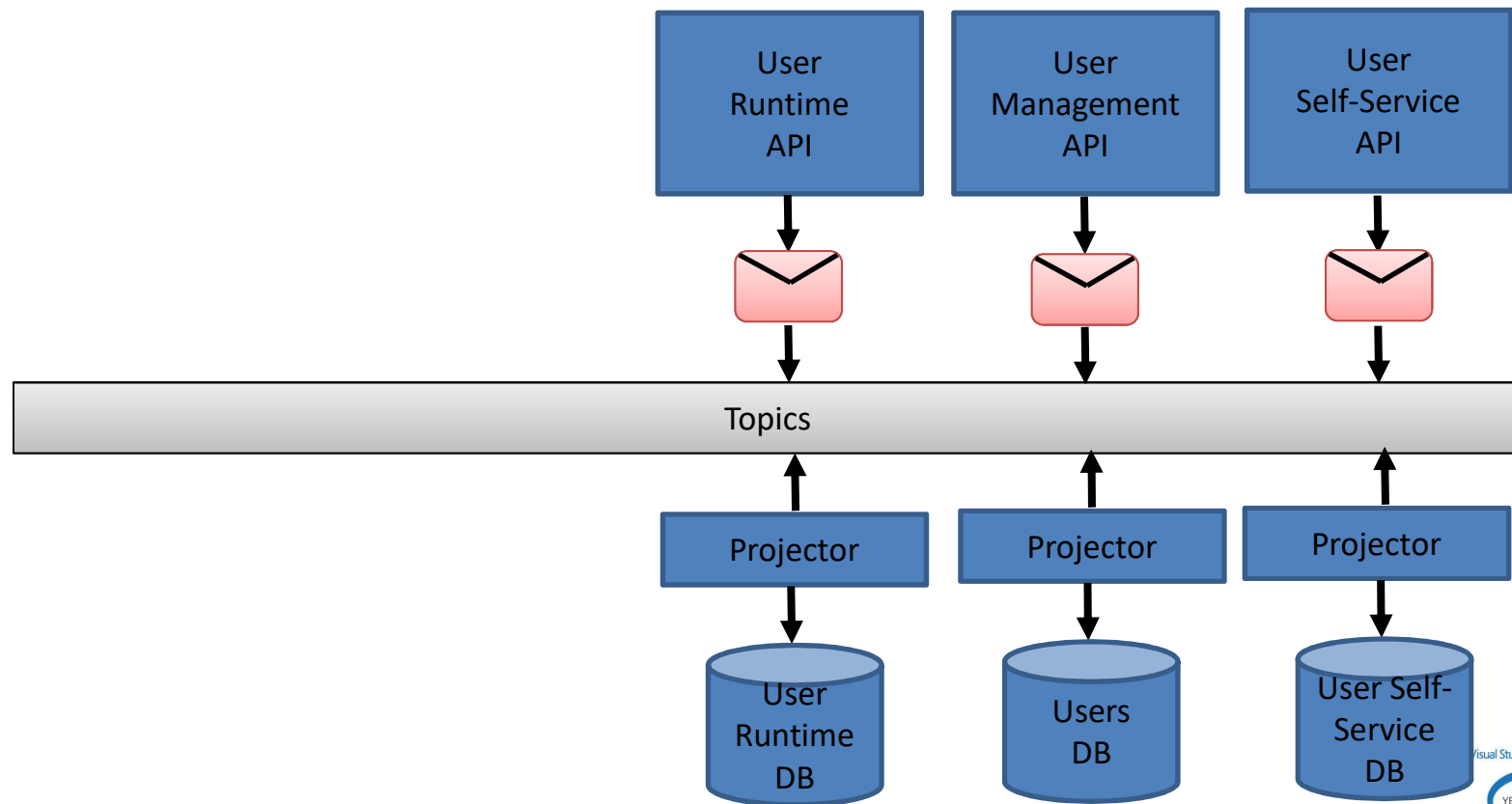
### End-to-end transaction

Operation ID: ea9581ac-4ea3372707edfa3d

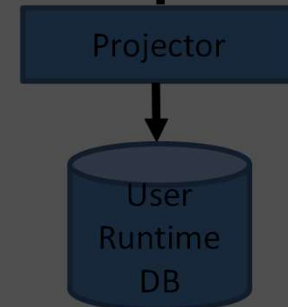
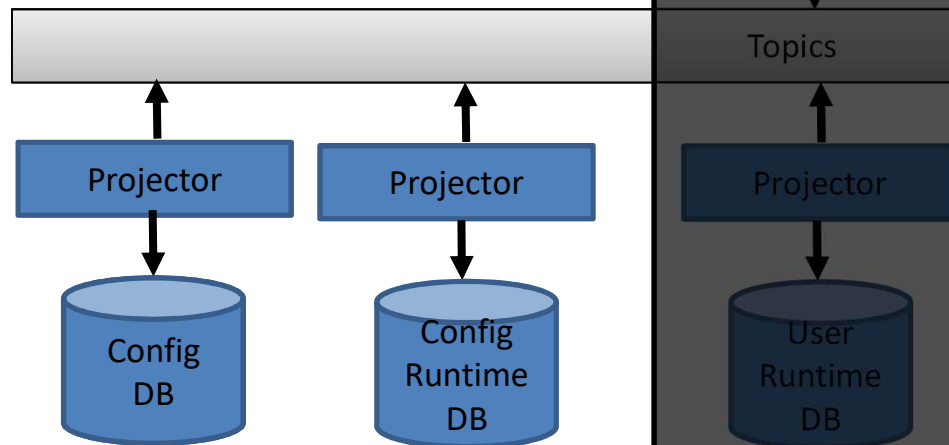
■ = Request (incoming) ■ = Dependency (outgoing) ▲ = Exception ▨ = Profiler trace □ = Debug snapshot

EVENT	RES.	DURATION	
✓ sm-signup.duckdns.org POST Account/EmailConfirm	500	123.5 ms	
✓ identityuser.profile.marathon.l4lb.thisdcos.directory POST /api/signup/email-verify	200	67 ms	
> identityuser.profile.marathon.l4lb.thisdcos.directory:5006 POST Signup/EmailVerify	200	62.6 ms	
sm-demos identityuser-dcos		6 ms	
sm-demos identityuser-dcos		2 ms	
sm-demos identityuser-dcos		1 ms	
sm-demos identityuser-dcos		7 ms	
▲ EXCEPTION System.NullReferenceException			

# History / Audit



# History / A

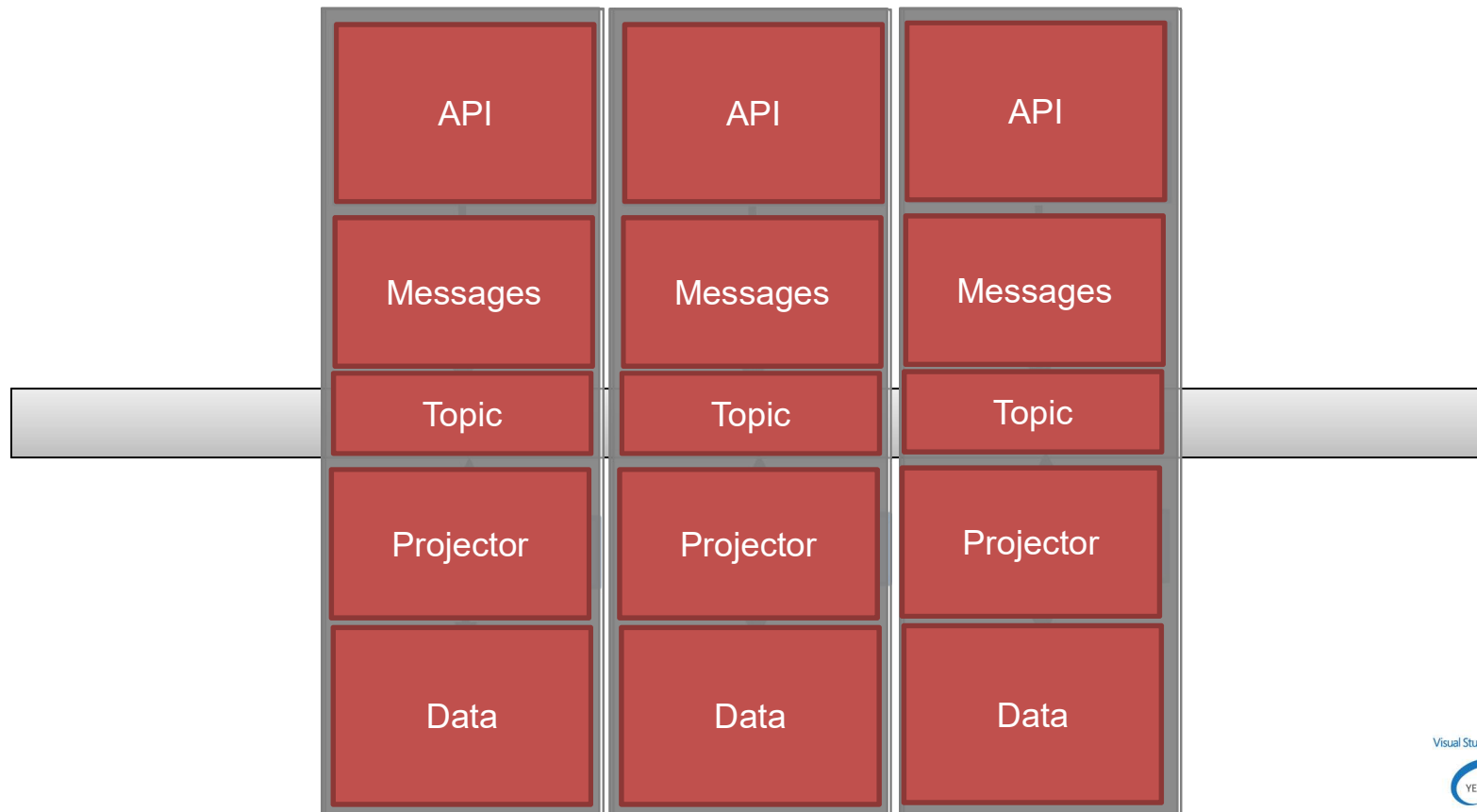


```
{
  {
    "id": "fd5b1598-f1d5-4282-9965-d6ed4e002ba3",
    "UserName": "ducky",
    "__messageType": "UserCreated"
  },
  {
    "id": "2d1c14b0-8973-407c-8ea5-d920ba930afe",
    "UserName": "ducky",
    "__messageType": "EmailConfirmed"
  },
  {
    "id": "5931c660-e91c-41d2-84a4-8f8ea3608b4a",
    "UserName": "ducky",
    "__messageType": "UserLoggedIn"
  },
  {
    "id": "6a75ee38-9355-4548-9462-70b8a4ec0d07",
    "UserName": "ducky",
    "__messageType": "UserLoggedOut"
  },
  {
    "id": "b8f563cf-0555-4151-8c58-36ba72e3bbc6",
    "UserName": "ducky",
    "__messageType": "UserLoginFailed"
  },
  {
    "id": "2334a700-69c3-4de4-9572-7782a790c526",
    "UserName": "ducky",
    "__messageType": "UserLockedOut"
  },
  {
    "id": "12809b61-dc76-486d-ae39-b5a364e22d2f",
    "UserName": "ducky",
    "__messageType": "UserUnlocked"
  },
}
```

Producing Audit and History from Messages

**DEMO**

# Manageability





# Benefits

- Focus on microservices owning their own data
- Build new projections for history tables, audit tables, aggregation
- Capture all events, replay from the beginning of time
  - Or, from snapshots
- Gather critical end to end tracing, forensic data including message flow

**AT FIRST, THIS WILL BE MORE WORK...**



# Resources

- We produced this container lab for Microsoft workshops walking you through Docker and AKS setup and deployments including load balancing:
  - <https://github.com/Microsoft/MCW-Containers-and-DevOps>
- We also helped with many of these other MCW labs:
  - <https://github.com/Microsoft?utf8=%E2%9C%93&q=mcw>