LIVE! 360
TECH EVENTS WITH PERSPECTIVE

Visual Studio LIVE! | SQL Server LIVE! | TECHMENTOR | Artificial Intelligence LIVE! | Office & SharePoint LIVE! | Modern Apps LIVE!

Code First - Database First

Where are you on the spectrum and why?

Tony Sellars
Lead Consultant,
Magenic

Level: Intermediate

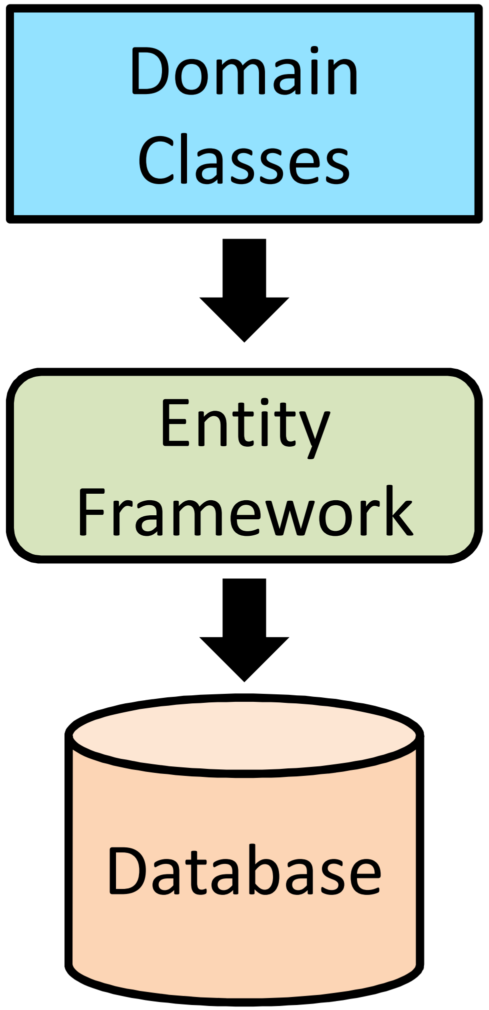
The Ultimate Education Destination
2018 Orlando

Ground Rules

- There are no silver bullets
- SQL Server
- Entity Framework / EF Core

What is Code First?

- Write Data Model Entity Classes
- ORM Generates the Database
- Updates are transferred to the database through Migrations.

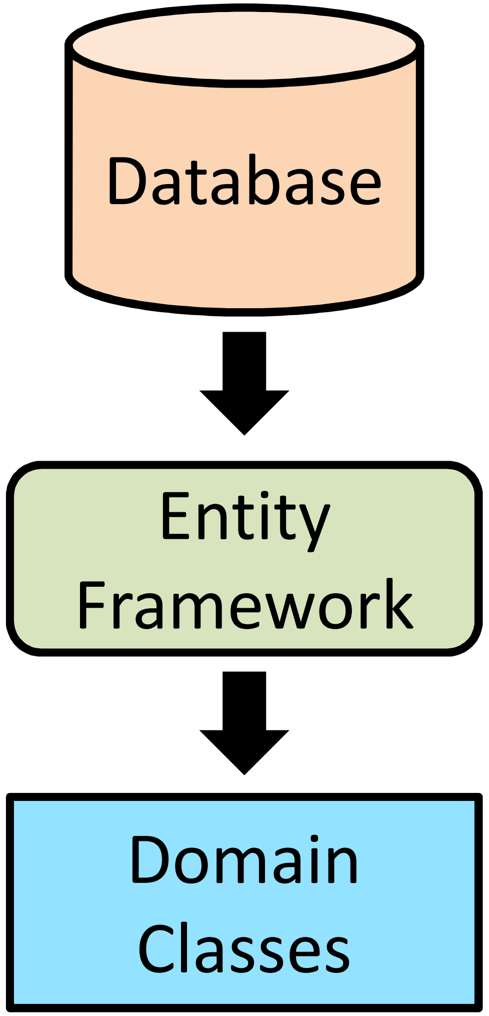


What do you get?

- Initial Development is very straight forward
- Migration Script tracking
- Integrated deployments

What is Database First?

- Existing or New Database
- ORM Generates Domain Classes
- Database code and updates managed in separate project.
  - Visual Studio / Red Gate



What do you get?

- Large number of tooling options
- Highly integrated across object types
- Database and Server level

Five W’s

Why

Who?

- Team(s) involved

When?

- Product Life Cycle

Where?

- The environment

What?

- Characteristic of the Application

Who - Teams

- Development
  - Database Resource
- Support
  - Experience / Technical Level
  - Level of support
  - Processes in place

When – Product Life Cycle

- Proof of Concept
- Greenfield
- Addition/Adjustment to existing application
- New component within existing system
- Change in technology

Where –

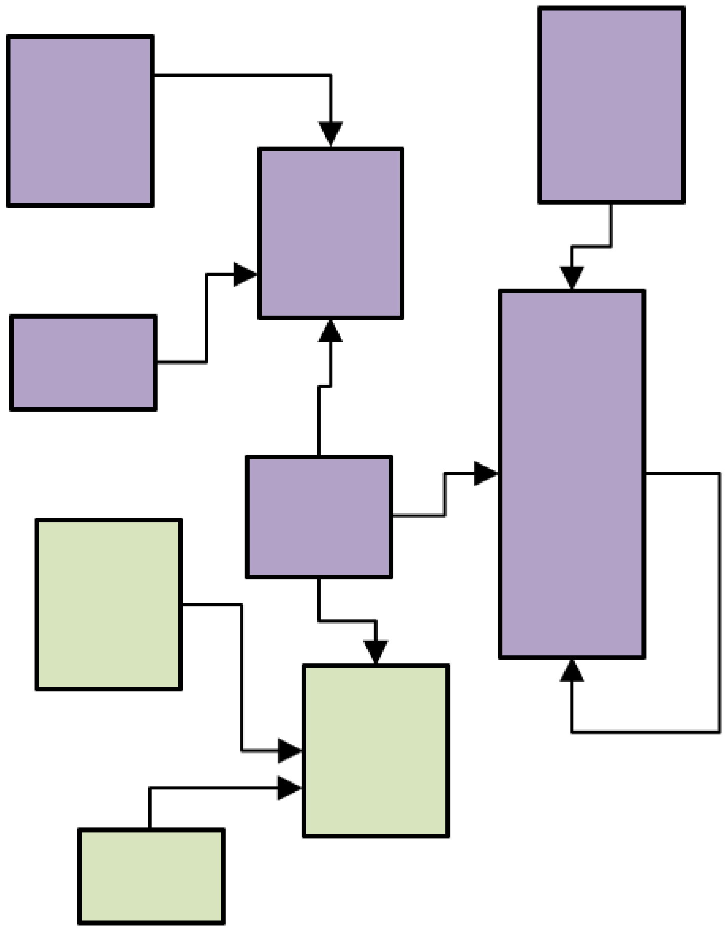
- Cloud / On Prem
- Complexity of the environment
- Stand alone or integrated
- What configurations or options does your application need.

What –

- Domain Space
- Data Profile
- Level of Interaction

Domain Space

- Number of Objects
- Size of Objects
- Complexity
- Using all or part of it .
  - Subset of the domain
  - Subset of objects themselves



Not all data is created equal

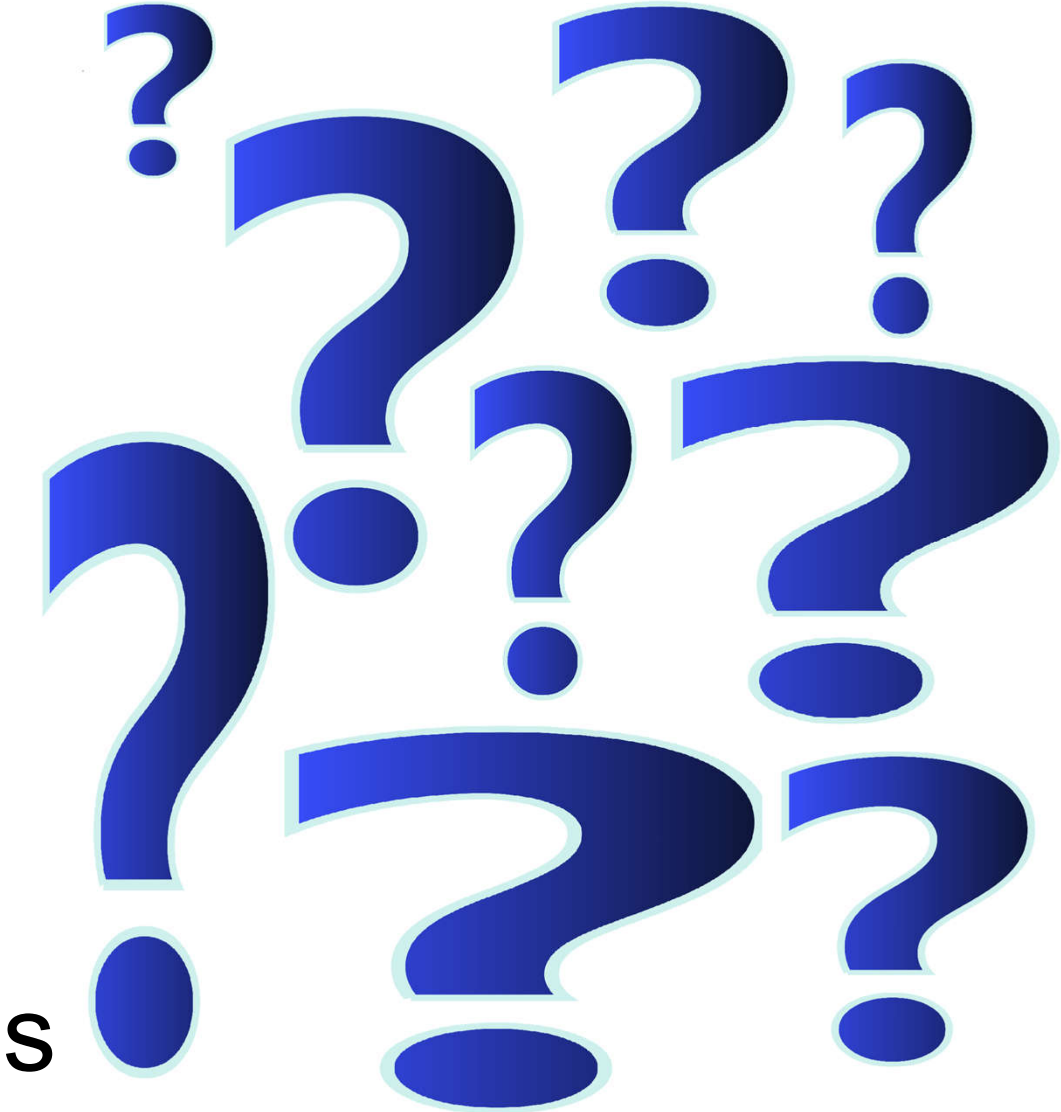
- Owner
- Consumer
- Life expectancy
- Volatility
- Consistency
- Location
- Security
- Volume

Level of Interaction

- Matching up
  - Object-Relational Impedance Mismatch
- Access Footprint
  - Just to see how the copy looks if it goes deep enough to reach the bottom.
    - You never know how much copy will be on a slide

Questions Overlap

- Size
  - The number of objects
- Complexity
  - The types of relationships
- Using all or part of it .
  - Subset of the domain
  - Subset of objects themselves



Use Case Samples

- Microservices
- Document Processing
- Document Review
- Shared Database
  - Direct Map
  - Abstraction Layer

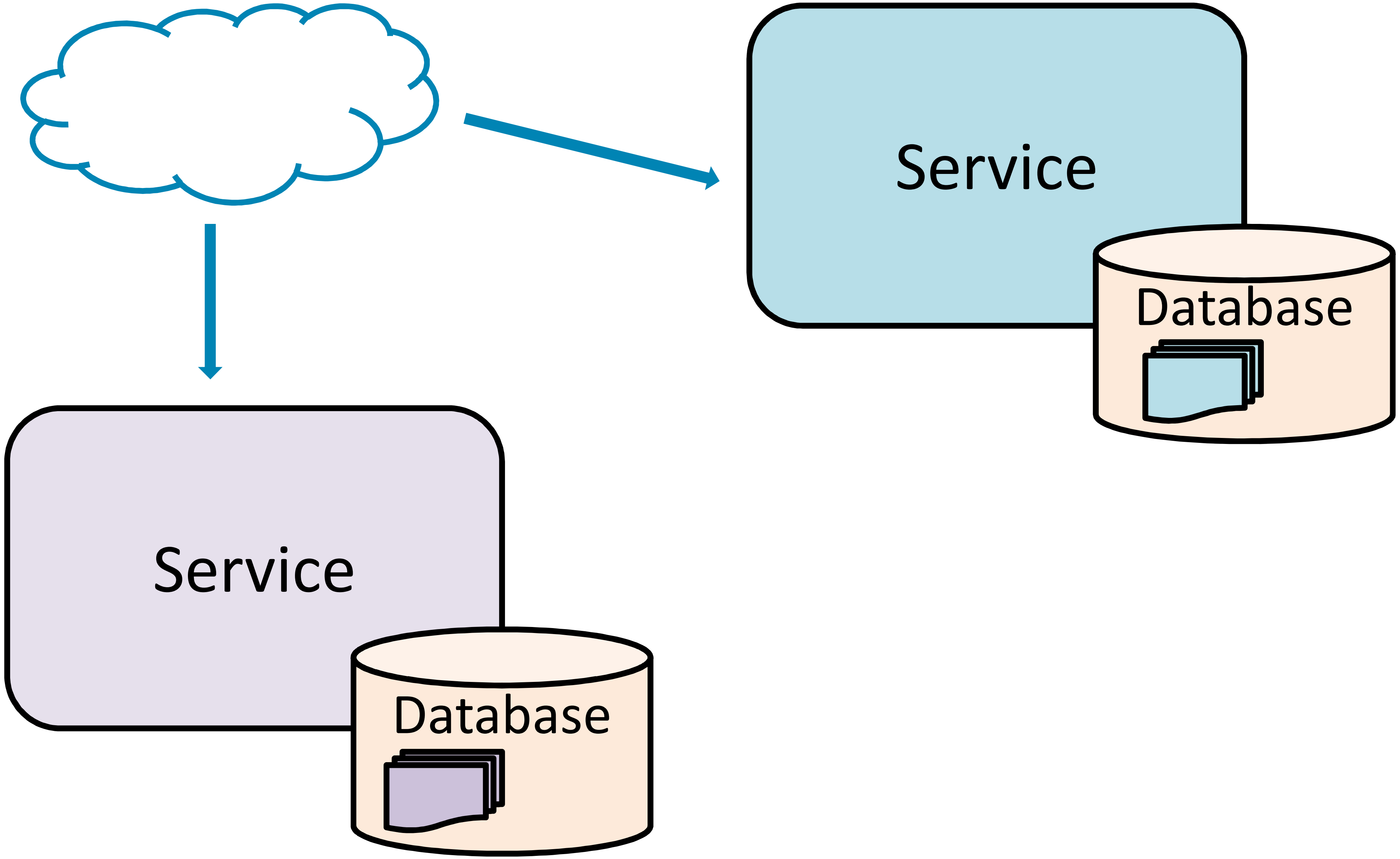
Microservices

- Dedicated Database
  - Isolated Data
- Dedicated Database
  - Shared Data
- Shared Database
  - Direct Map
  - Abstraction Layer

Dedicated w/ Isolated Data

- 1:1 mapping
- Application is sole consumer of data
- Small to Mid size schema
- Lower data volumes
- The classic case for Code First

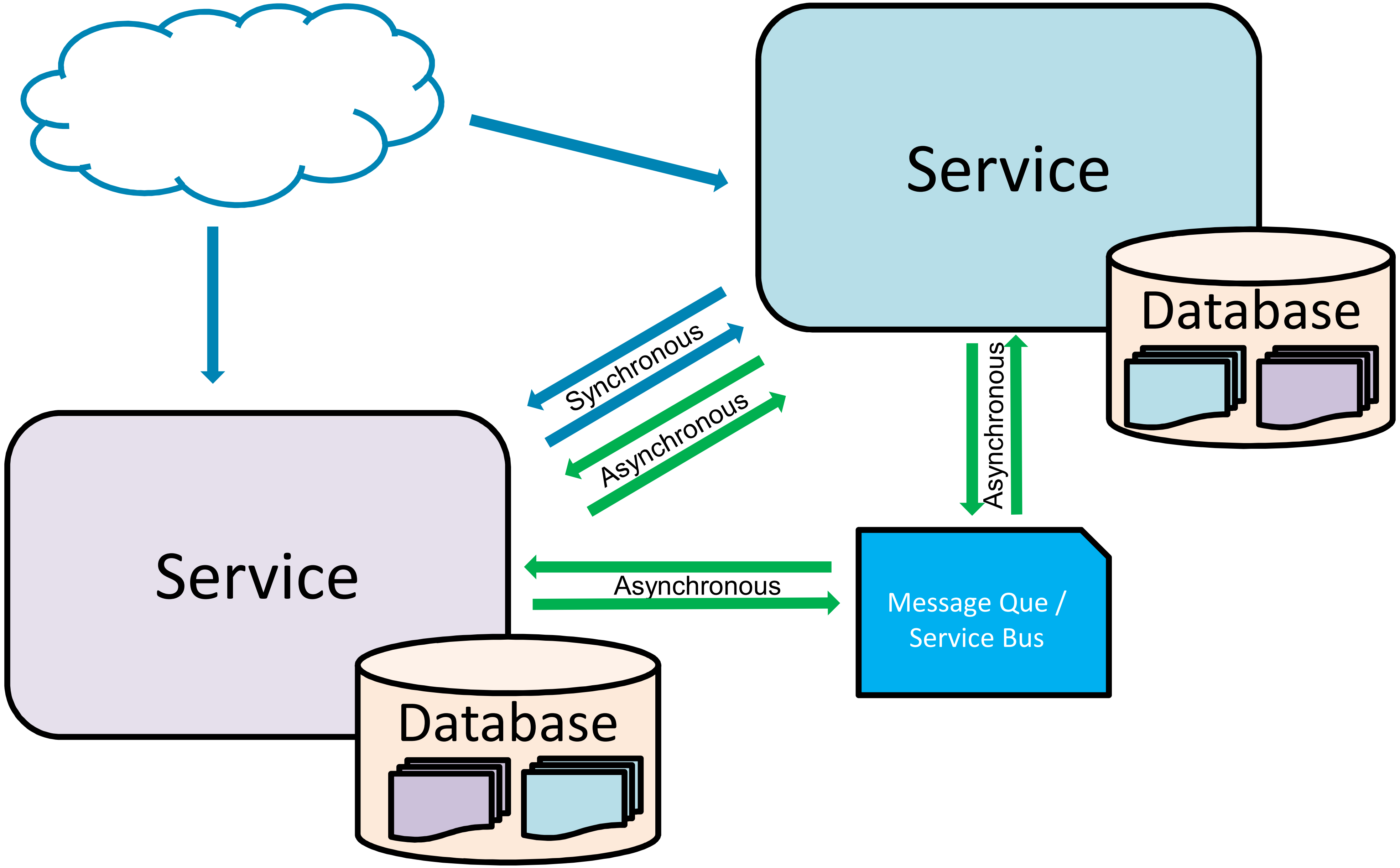
Isolated Data



Dedicated w/ Shared Data

- 1:1 mapping
- Model overlaps with other system
- Small to Mid size schema
- Lower data volumes
- Still a good case for Code First
- Syncing Considerations

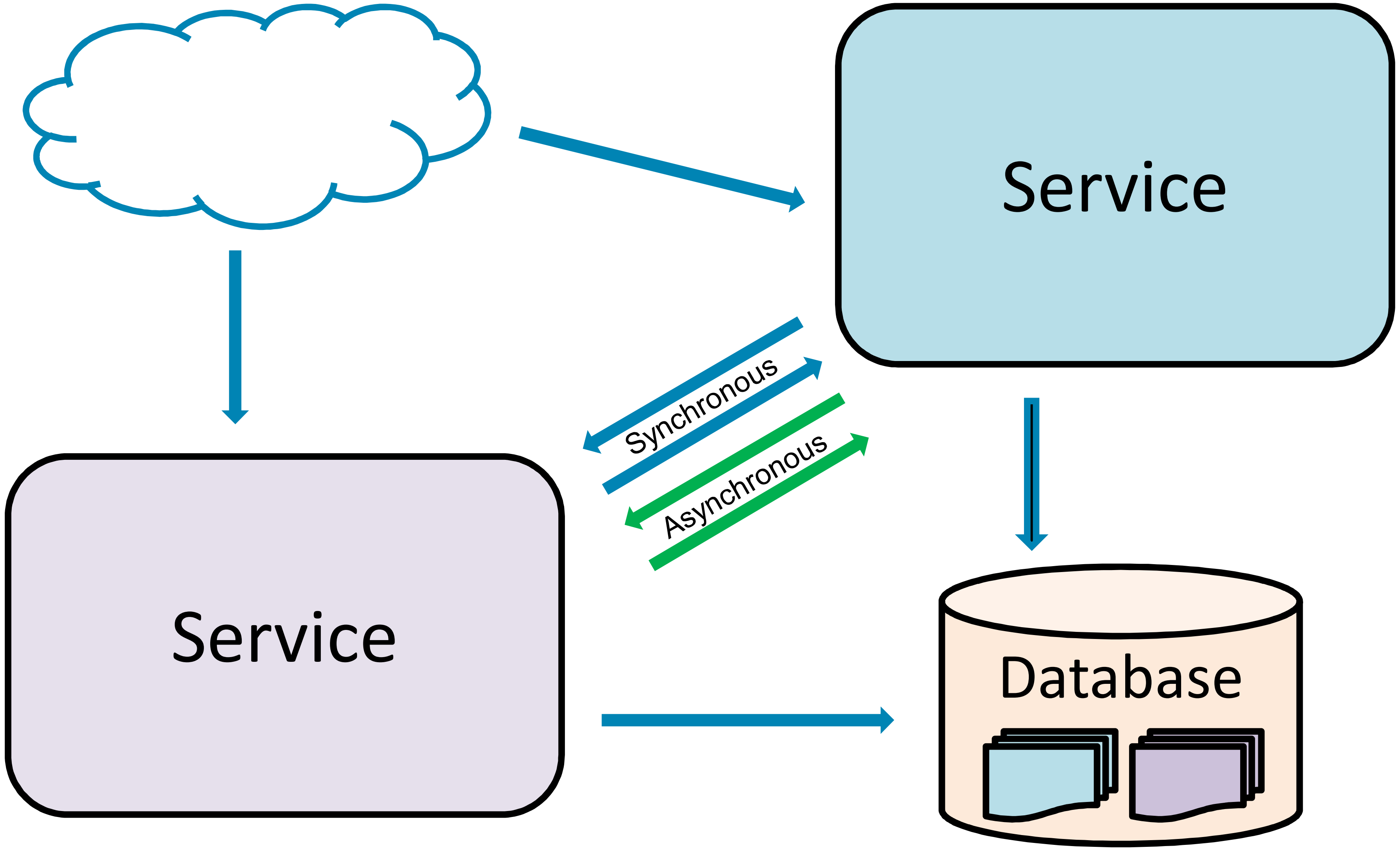
Shared Data


Service
Database
Synchronous
Asynchronous
Asynchronous
Service
Database
Asynchronous
Message Que /
Service Bus

Shared (Direct Map)

- Each application logically maps to subset of tables
- Mid to Large size schema
- Medium data volumes
- Leans towards the Database First side

Direct Map

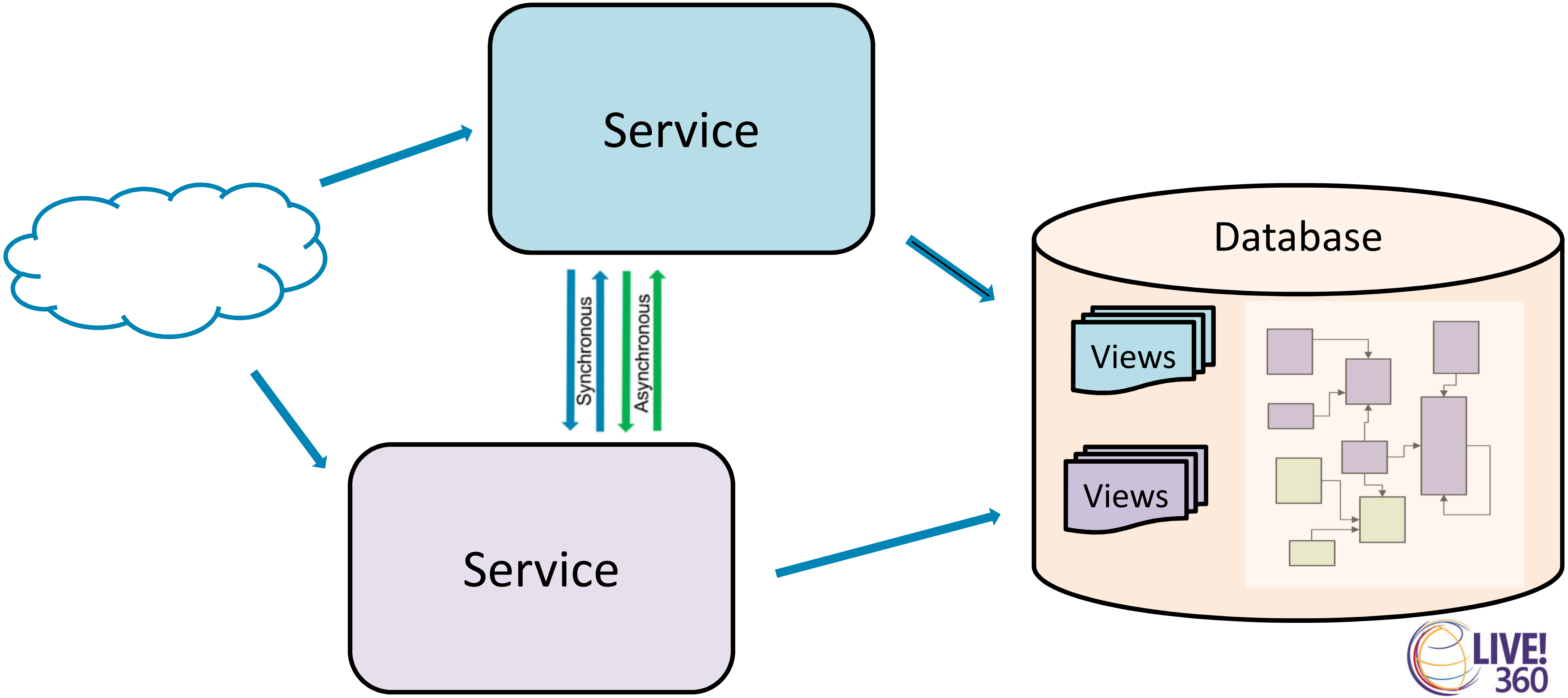


LIVE! 360
TECH EVENTS WITH PERSPECTIVE

Shared (Abstracted)

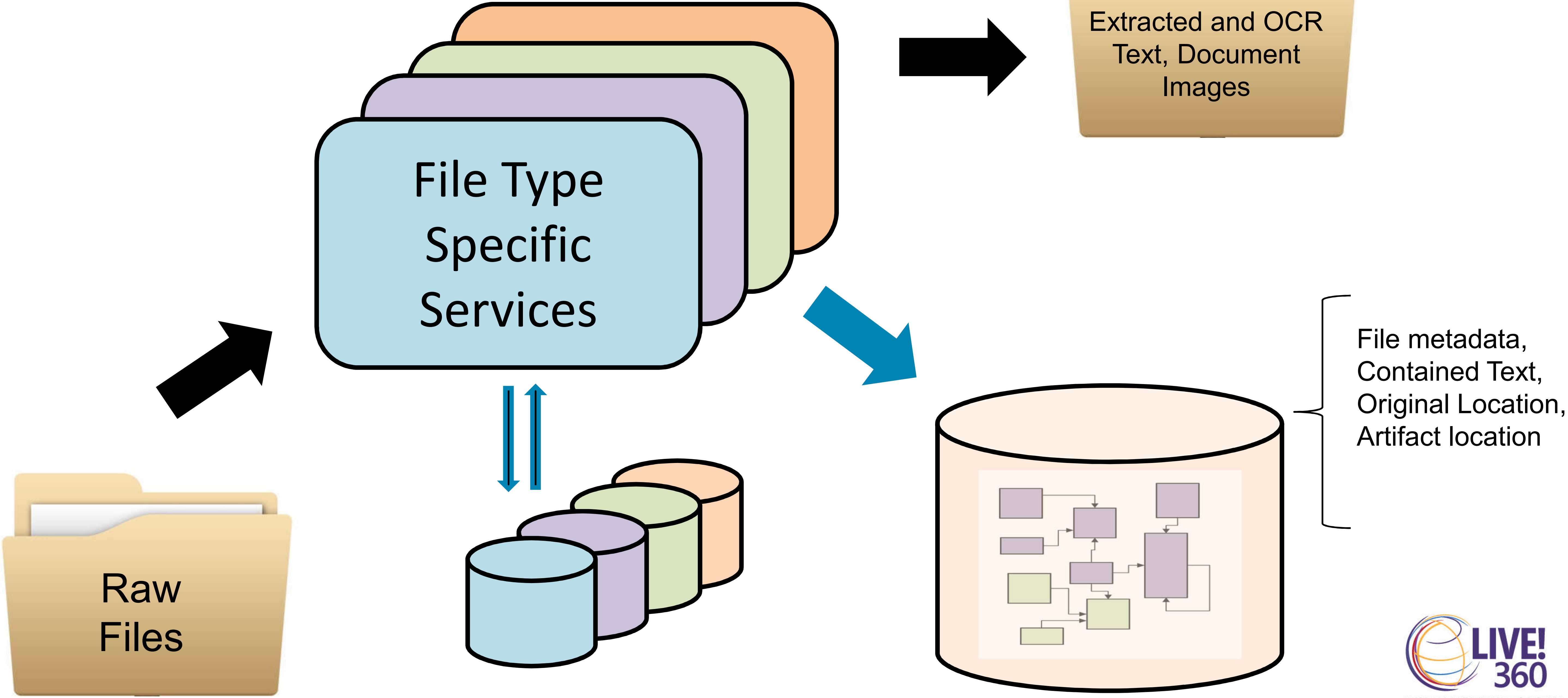
- Each application logically maps
  - Subset of tables
  - Portions of tables
  - Composite tables
- Size of the schema not relevant
- About any data volume
- Leans towards the Database First side

Abstracted



LIVE! 360
TECH EVENTS WITH PERSPECTIVE

Document Processing



Raw Files

File Type Specific Services

Extracted and OCR Text, Document Images

File metadata,
Contained Text,
Original Location,
Artifact location

What can you run into?

- Complex Queries
- Complex Migrations
- Maintenance
- Need more features

Complex Query

- ORM basically a SQL Code Generator
- Include()
  - Can be chained
  - Can be fastest way to complex code
- Impacted by volume

Mitigation

- Procedures (functions)
  - Parameterization, Temp Objects, etc.
  - Per call or per unit-of-work
- Views
  - Treated like table
- Indexing
  - Treated like table

```
public partial class spMyProcedure : Migration
{
   protected override void Up(MigrationBuilder migrationBuilder)
   {
      var sql = @"CREATE PROCEDURE [dbo].[MyProcedure]
            @Param1 int,
            @Param2 nvarchar(50)
         AS
         BEGIN
         SET NOCOUNT ON;
         SELECT TOP(@Param1)
         *
         FROM dbo.MyTable
         WHERE StringField like @Param2 +'%’
         END";
      migrationBuilder.Sql(sql);
   }

   protected override void Down(MigrationBuilder migrationBuilder)
   {

   }
}
```

Complex Migration

- Existing Data
- Add a Not Null Column
- Changes to Large tables
- Split or Combine tables

Maintenance

- Changes to Database Objects over time
- “Code” is a “String” field
- Overwrite previous migration definition with a new migration definition.

Need More Features

- Replication
- Row Level Security
- Dynamic Data Masking
- Partitioning
- Clustering/High Availability

Change your mind?

- Convert to Database First
  - Large schema growth
  - Volume of non-table object reaches critical mass
- Convert to Code First
  - Split off schema section from the overall
  - Schema never really grew as expected

Patterns

- Convert targeted problem queries to stored procedure calls.
- Convert all read operations to procedure calls.
  - Procedures

    Parameterization, Temp Objects, etc.
  - Views

    Treated like a table

Reporting and Export

- Maintain Relationships
  - May not be used in application logic but may be useful downstream.
- Coordinate Enumerations
  - Dedicated DB may not keep physical table

Contact Me

TonyS@Magenic.com
www.linkedin.com/in/tonysell/
Twitter: @RA_TonySellars
Blog: wordpress.com/view/tonysellars.wordpress.com