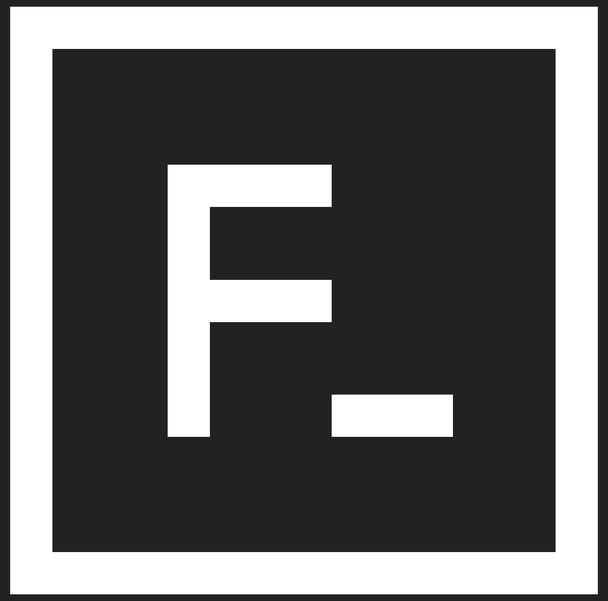
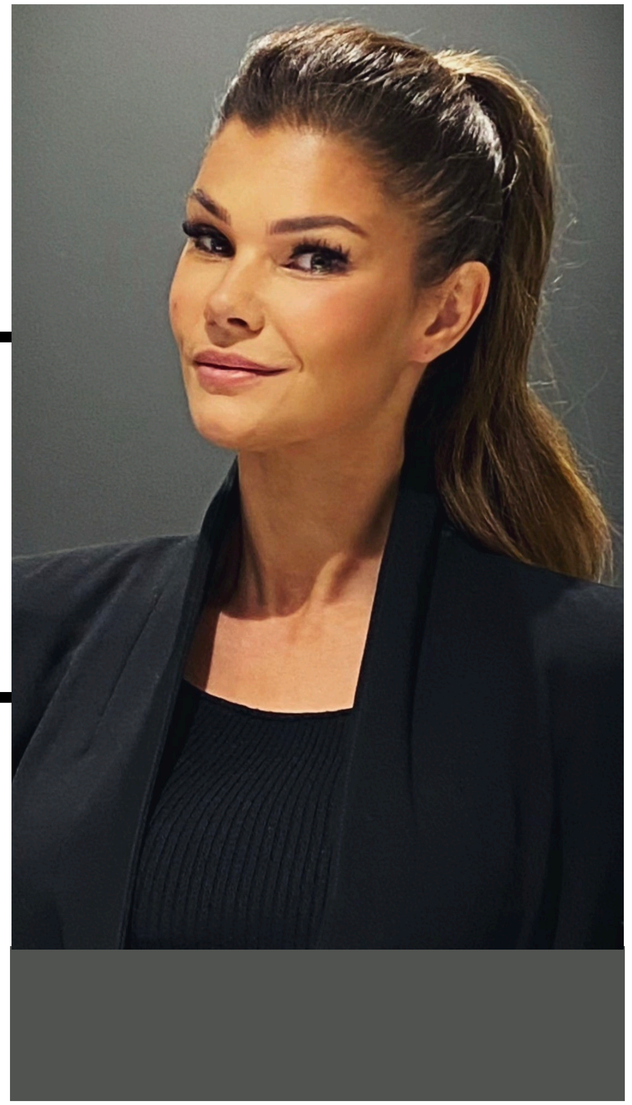
.NET & AZURE COMMUNITY - FORTE DIGITAL
PRESENTATION BY MONICA BEATE TVEDT, DIRECTOR OF TECHNOLOGY





Monica Beate Tvedt

Teknologidirektør

Forte_ Digital

TIDLIGERE

- Agency Director - Head of Microsoft Development, Mixed Reality & Microservices at Sopra Steria
- Head of UMS Innovation Center at Unified Messaging Systems
- Global Head of SaaS Development at Unified Messaging Systems
- Senior Software Engineer Consultant, Webstep @ Sparebanken Vest
- Software Engineer, CellVision
- Gründer

PROSJEKT 2020

- Kunde: **ASKO**
  Rolle: Arkitekt og Front-end lead
- Kunde: **Kværner**
  Rolle: Arkitekt og Mobilspesialist
- Kunde: **COVID-19 Digital Feberpoliklinikk**
  Rolle**:** Løsningsarkitekt

FOREDRAG 2020

Oslo Business Forum 2020, Relevans 2020, Global AI on Tour 2020, Women in Tech 2020, Lørn.Tech.

DIVERSE NTERESSER

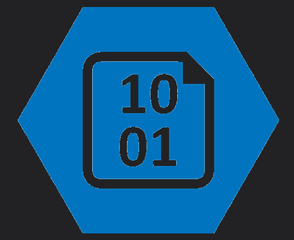
Alpint, tennis, programmering, tegne, lese bøker

0.0	Recap: Events & Messages
1.0	Choosing a data storage approach
2.0	Azure Storage Account
3.0	Connecting your App to Blob Storage
4.0	Self Study - Storage Security & Azure Defender
-	-
5.0	In-depth Self Study

0.0 - Recap

What we learned at our last meetup.
Messages & Events..

- **Azure Storage queues**
- **Azure Event Hubs**
- **Azure Event Grid**
- **Azure Service Bus**



1.0
Choosing a data storage approach.

Focusing on Azure Storage, Azure SQL Database, and Azure Cosmos DB

Storing your data in Azure

We will learn how to classify our data as structured, semi-structured, or unstructured - and how using Azure Storage, Azure SQL Database, and Azure Cosmos DB - or a combination of them - is the best way to get the most performant solution for your business scenario.

Azure provides a variety of ways to store data:

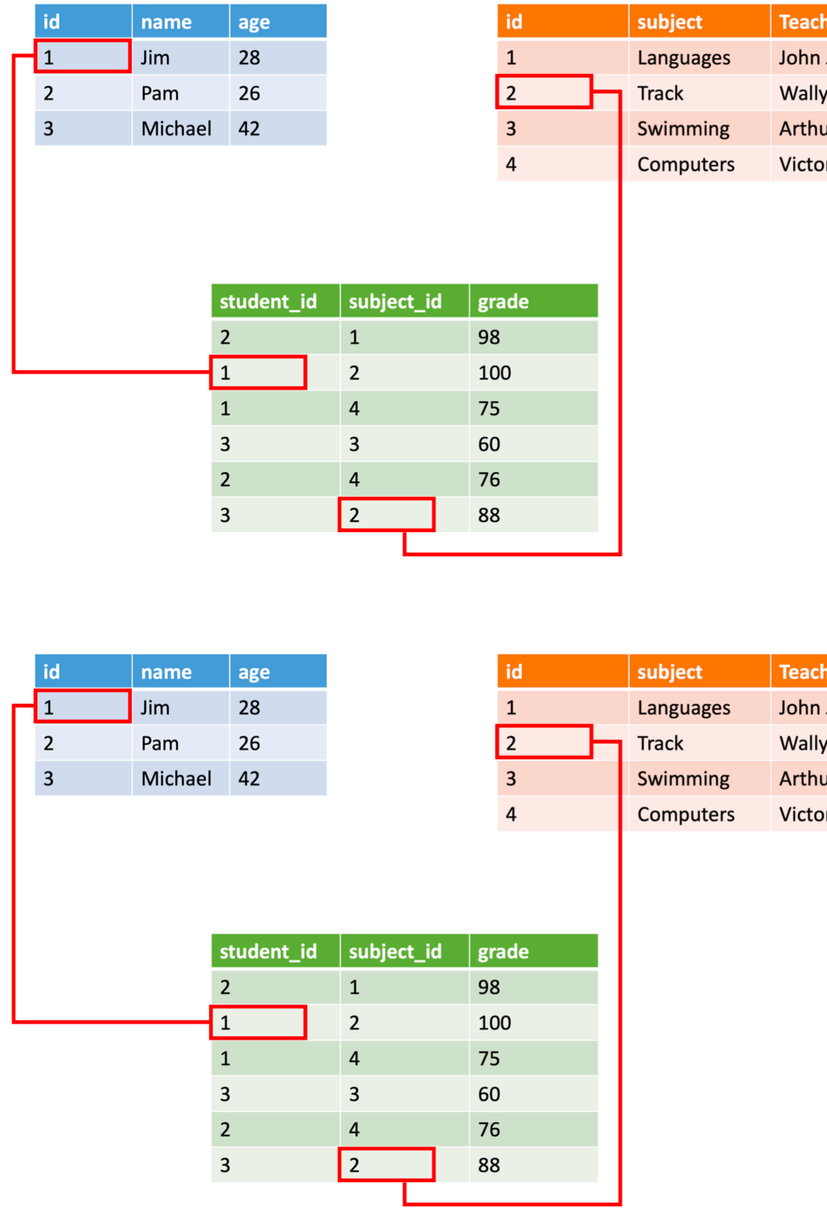
- **Unstructured**
- **Archival**
- **Relational**
- **and more..**


MENY
SUNN-
SPIRASJON
Sunn inspirasjon til alle måltider
Oppskrifter og enkle tips →

Numero Uno: Classifying your Data

Structured data, sometimes referred to as *relational data*, is data that adheres to a strict schema, so all of the data has the same fields or properties. The shared schema allows this type of data to be easily searched with query languages such as SQL (Structured Query Language). This capability makes this data style perfect for applications such as **CRM systems, reservations, and inventory management**.

It’s easy to enter, query, and analyze.



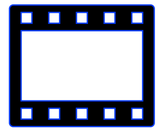
Semi-Structured Data

XML, JSON, and XAML.

Semi-structured data is less organized than structured data, and is not stored in a relational format, as the fields do not neatly fit into tables, rows, and columns. Semi-structured data contains tags that make the organization and hierarchy of the data apparent - for example, key/value pairs.

```
<Person Age="23">
   <FirstName>John</FirstName>
   <LastName>Smith</LastName>
   <Hobbies>
      <Hobby Type="Sports">Golf</
      <Hobby Type="Leisure">Read
      <Hobby Type="Leisure">Guita
   </Hobbies>
</Person>
```

```
{
   "firstName": "John",
   "lastName": "Doe",
   "age": "23",
   "hobbies": [
      { "type": "Sports", "value": "Gol
      { "type": "Leisure", "value": "Re
      { "type": "Leisure", "value": "Gu
   ]
}
```



Unstructured Data

The organization of unstructured data is ambiguous. Unstructured data is often delivered in files, such as photos or videos. The video file itself may have an overall structure and come with semi-structured metadata, but the data that comprises the video itself is unstructured. Therefore, photos, videos, and other similar files are classified as unstructured data.

Examples of unstructured data include:

- Media files, such as photos, videos, and audio files
- Office files, such as Word documents
- Text files
- Log files

Product Data

Groceries are fairly structured in nature, as each product has a product SKU (bar code), a description, a quantity, a price etc.

However, as you introduce new products or different kinds of products, you may want to add **different fields** as time goes on. What if a new product introduces a “CO_2 impact”-value, and you want to enable customers to filter on “environment friendly" groceries in the future.

How does this impact your data?
Data classification?

Photos & Videos

The photos and videos displayed on a product page or a recipe page are unstructured data. Although the media file may contain metadata, the body of the media file is unstructured.

Data classification: **Unstructured**

Business Data

Business analysts want to implement business intelligence to perform inventory pipeline evaluations and sales data reviews. In order to perform these operations, data from multiple months needs to be aggregated together, and then queried

Data classification?

Summary

- *Structured data* is organized data that neatly fits into rows and columns in tables.
- *Semi-structured data* is still organized and has clear properties and values, but there's variety to the data.
- *Unstructured data* doesn't fit neatly into tables, nor does it have a schema.

Knowledge Check

- A **JSON file** is an example of which type of data?

- A **video** is an example of which type of data?

Dos: Determine operational needs

When deciding what storage solution to use, think about how your data will be used. How often will your data be accessed? Is your data read-only? Does query time matter? The answers to these questions will help you decide on the best storage solution for your data. At MENY our customers need quick access to product data, and business users need to run complex analytical queries.

Ask yourself these questions when determine operational needs:

- Will you be doing simple lookups using an ID?
- Do you need to query the database for one or more fields?
- How many create, update, and delete operations do you expect?
- Do you need to run complex analytical queries?
- How quickly do these operations need to complete?

Product Data

- Customers will want to query the product catalog (groceries, recipes).
- Customer needs may require lots of read operations, with the ability to query on certain fields.
- When customers place orders, the application must update product quantities.
- The update operations need to happen just as quickly as the read operations so that users don't put an item in their shopping carts when that item has just sold out.

Conclusion: large number of read operations, but will also require increased write operations.

Photos & Videos

The photos and videos that are displayed on product pages have different requirements. They need fast retrieval times so that they are displayed on the site at the same time as product catalog data, but they don't need to be queried independently. Instead, you can rely on the results of the product query, and include the video ID or URL as a property on the product data. So, photos and videos need only be retrieved by their ID.

Users will not make updates to existing photos or videos

Conclusion: Queried by ID to return the whole file. Write operations are less of a priority.

Business Data

- Analysis is happening on historical data.
- No original data is updated based on the analysis, so business data is read-only.
- Users don't expect their complex analytics to run instantly, some latency in the results is okay.
- Business data will be stored in multiple data sets, but business analysts will be able to read from all databases.

Conclusion: Read-only historical data with some expected latency.

Summary

When deciding what storage solution to use, think about how your data will be used. How ***often*** will your data be ***accessed***? Is your data ***read-only***? Does ***query time*** matter?

The answers to these questions will help you decide on the best storage solution for your data.

Tres: Transactions

Applications may need to group a series of data updates together, because a change to one piece of data needs to result in a change to another piece of data. Transactions enable you to group these updates so that if one event in a series of updates fails, the entire series can be rolled back, or undone.

At MENY for which Scenarios would this be applicable?

What is a transaction?

A transaction is a logical group of database operations that execute together.
Will a change to one piece of data in your dataset impact another? If the answer is yes, then you'll need support for transactions in your database service.

Transactions are often defined by a set of four requirements, referred to as ACID guarantees:

- **Atomicity** means a transaction must execute exactly once and must be atomic; either all of the work is done, or none of it is. Operations within a transaction usually share a common intent and are interdependent.
- **Consistency** ensures that the data is consistent both before and after the transaction.
- **Isolation** ensures that one transaction is not impacted by another transaction.
- **Durability** means that the changes made due to the transaction are permanently saved in the system. Committed data is saved by the system so that even in the event of a failure and system restart, the data is available in its correct state.

OLTP vs OLAP

Transactional databases are often called **OLTP** (Online Transaction Processing) systems. OLTP systems commonly support lots of users, have quick response times, and handle large volumes of data. They are also highly available (meaning they have very minimal downtime), and typically handle small or relatively simple transactions.

On the contrary, **OLAP** (Online Analytical Processing) systems commonly support fewer users, have longer response times, can be less available, and typically handle large and complex transactions

Product Data

Product catalog data should be stored in a transactional database. When users place an order and the payment is verified, the inventory for the item should be updated. Likewise, if the customer's credit card is declined, the order should be rolled back, and the inventory should not be updated. These relationships all require transactions.

Conclusion: Transactional Database.

Photos & Videos

Photos and videos in a product catalog don't require transactional support. These files are changed only when an update is made or new files are added by the Administrator. Even though there is a relationship between the image and the actual product data, it's not transactional in nature

Conclusion: No need for a Transactional Database.

Business Data

- Analysis is happening on historical data.
- No original data is updated based on the analysis, so business data is read-only.

Question: Do we need transactional support?

Knowledge Check

1. Which type of transactional database system would work best for product data?
   OLAP *or* **OLTP**

2. Suppose the operations to **update inventory** and **process payments**
   are in the **same transaction**. A user is attempting to apply a $30 store credit for the full amount of an order, and submitted the exact same order using the store credit (for the full amount) using their phone and laptop at the same time - so **two identical orders are received**.

   The database behind the scenes is an **ACID-compliant database**, what would happen?
   - Both orders would be processed and use the in-store credit.
   - One order would be processed and use the in-store credit, and the other order would update the remaining inventory for the items in the basket, but would not complete the order.
   - One order would be processed and use the in-store credit, and the other order would not be processed.

Choosing a data storage solution

Choosing the correct storage solution can lead to better performance, cost savings, and improved manageability. Here, we'll apply what we've learned about the data at **MENY**, and find the best Azure service for each data set.

Product Data Summary

Data classification: Semi-structured because of the need to extend or modify the schema for new products.

Operations:

- Customers require a high number of read operations, with the ability to query many fields within the database.
- The business requires a high number of write operations to track its constantly changing inventory.

Latency & throughput: High throughput and low latency.
Transactional support: Because all of the data is both historical and yet changing, transactional support is required.

Recommended service:
Azure Cosmos DB

(best suited for Product catalogs, Content management, and Inventory management)

- Azure Cosmos DB supports semi-structured data, or NoSQL data, by design. So, supporting new fields, such as the **“CO_2 impact”-field** or any new fields you need in the future, is a given with Azure Cosmos DB.
- Azure Cosmos DB supports SQL for queries and **every property is indexed by default**. You can create queries so that your customers can filter on any property in the catalog.
- Azure Cosmos DB is also **ACID-compliant**, so you can be assured that your transactions are completed according to those strict requirements.
- You can **scale** up to handle higher customer demand during peak shopping times, or scale down during slower times to conserve cost.

Key	Document
1001	{ “CustomerID”: 99, “OrderItems”: [{ “ProductID”: 2010, “Quantity”: 2, “Cost”: 520 }, { “ProductID”: 4365, “Quantity”: 1, “Cost”: 18 }], “OrderDate”: “04/01/2017” }
1002	{ “CustomerID”: 220, “OrderItems”: [{ “ProductID”: 1285, “Quantity”: 1, “Cost”: 120 }], “OrderDate”: “05/08/2017” }

As an added plus, Azure Cosmos DB also enables you to replicate your data anywhere in the world with the click of a button. So, if your e-commerce site has users concentrated in the US, France, and England, you can replicate your data to those data centers to **reduce latency**, as you've physically moved the data closer to your users.

Why not **Azure SQL Database?**

- Would be an excellent choice for this data set if you could identify the subset of properties that are **common for most of the products** and the variable properties that might not exist in some products.
- Azure SQL Database enables you to combine structured data in the columns, and **semi-structured data stored as JSON columns** that can be easily extended.
- Azure SQL Database can provide **many of the same benefits** of Azure Cosmos DB, but it provides little benefit if the structure of your data is changing in different entities, and you cannot pre-define a set of common properties that are repeated in most of the entities.
- Unlike Azure CosmosDB that index every property in the documents, in Azure SQL Database you **need to explicitly define what properties from semi-structured documents should be indexed**.

Azure Cosmos DB is better choice for highly unstructured and variable data where you cannot predict what are the properties that should be indexed.

Photos & Videos Summary

Data classification: Unstructured.

Operations:

- Only need to be retrieved by ID.
- Customers require a high number of read operations with low latency.
- Creates and updates will be somewhat infrequent and can have higher latency than read operations.

Latency & throughput: Retrievals by ID need to support low latency and high throughput.
Creates and updates can have higher latency than read operations.
Transactional support: Not required.

Recommended service:
Azure Blob storage

- Azure Blob storage supports storing files such as photos and videos. It also works with Azure Content Delivery Network (CDN) by caching the most frequently used content and storing it on edge servers. Azure CDN reduces latency in serving up those images to your users.

- By using Azure Blob storage, you can also move images from the hot storage tier to the cool or archive storage tier, to reduce costs and focus throughput on the most frequently viewed images and videos.

- Examples when Azure Blob Storage should be used: Images, videos, office documents, PDFs, Static HTML, JSON, CSS, Log and audit files, Database backups.

Business Data Summary

Data classification: Structured.

Operations: Read-only, complex analytical queries across multiple databases.

Latency & throughput: Some latency in the results is expected based on the complex nature of the queries.
Transactional support: Not required.

Recommended service:
Azure SQL Database

Business data will most likely be queried by business analysts, who are more likely to know SQL than any other query language.

- Azure SQL Database could be used as the solution by itself, but pairing it with **Azure Analysis Services** enables data analysts to create a semantic model over the data in SQL Database. The data analysts can then share it with business users, so that they only need to connect to the model from any **business intelligence (BI) tool** to immediately explore the data and gain insights.

Best suited for highly normalized data such as Inventory management, Order management, Reporting database, Accounting.

Why not **other services?**

- *Azure Synapse* supports OLAP solutions and SQL queries. But your business analysts will need to perform **cross-database queries**, which **Azure Synapse does not support**.

- *Azure Analysis Services* could be used in addition to Azure SQL Database. But your business analysts are more well-versed in SQL than in working with Power BI. So they'd like a database that **supports SQL queries**, which **Azure Analysis Services does not**.

  In addition, the financial data you're storing in your business data set is relational and multidimensional in nature. Azure Analysis Services supports tabular data stored on the service itself, but **not multidimensional data**. To analyze multidimensional data with Azure Analysis Services, you can use a direct query to the SQL Database.

- *Azure Stream Analytics* is a great way to analyze data and transform it into actionable insights, but its focus is on **real-time data** that is streaming in. In this scenario, the business analysts are **looking at historical data only.**

2.0
Azure Storage Accounts.

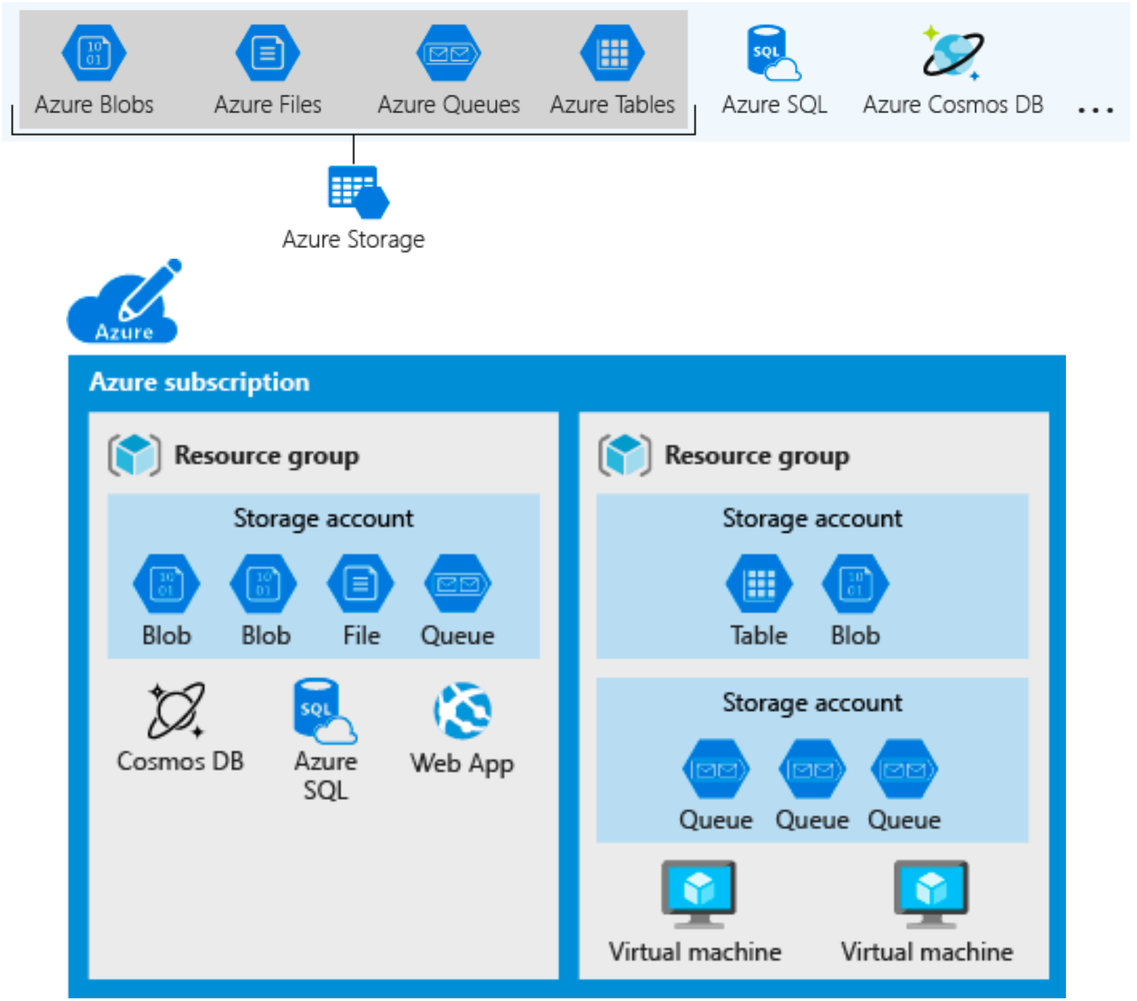
- Decide how many storage accounts you need for your project.
- Determine the appropriate settings for each storage account.

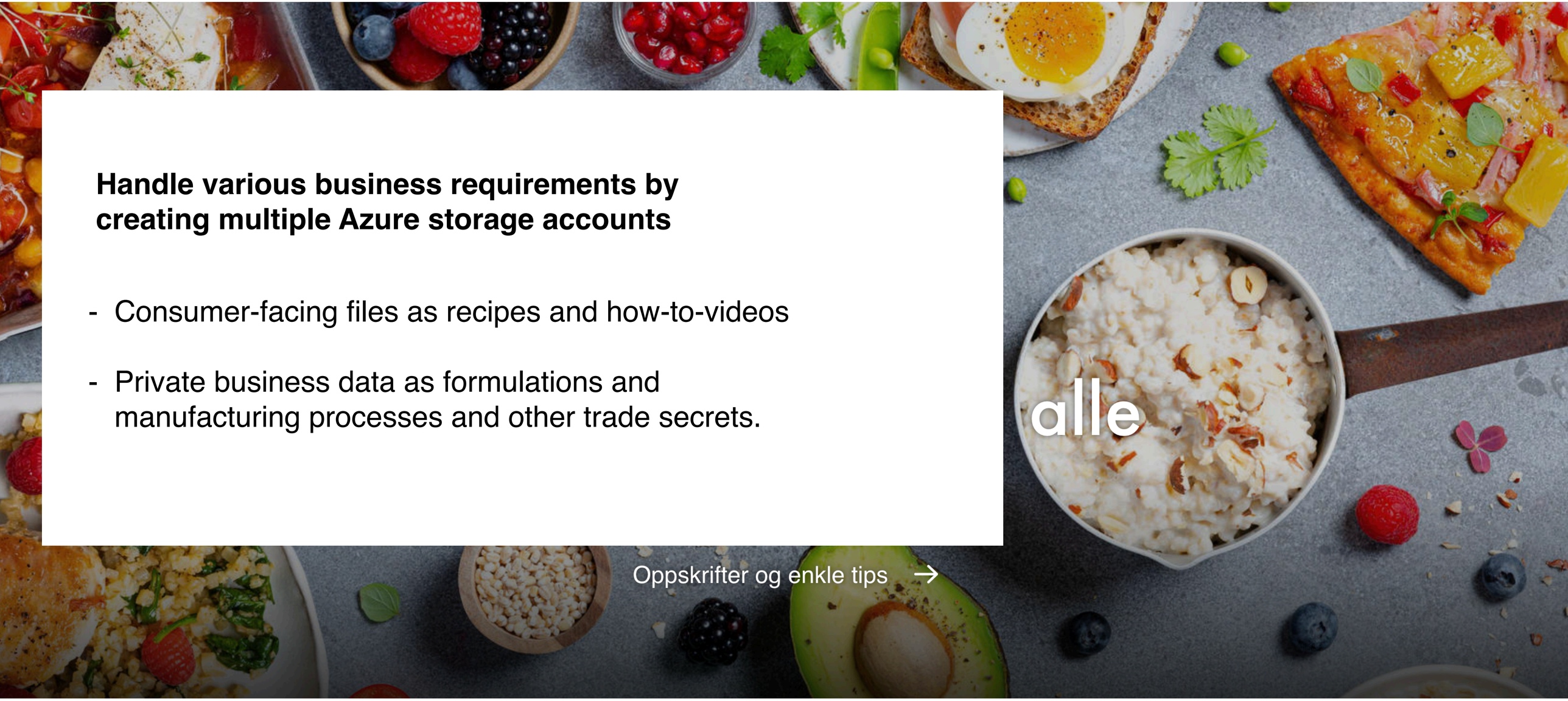
Azure Storage

Azure selected four of its data services and placed them together under the name Azure Storage.
The four services are *Azure Blobs, Azure Files, Azure Queues, and Azure Tables*.

These four were given special treatment because they are all primitive, cloud-based storage services and are often used together in the same application.

A storage account is an Azure resource and is included in a resource group.





Handle various business requirements by creating multiple Azure storage accounts

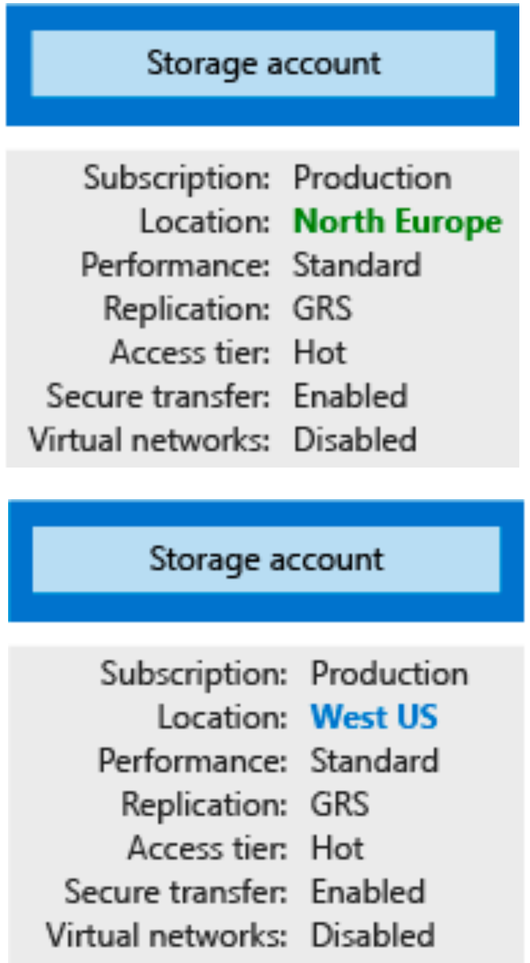
- Consumer-facing files as recipes and how-to-videos
- Private business data as formulations and manufacturing processes and other trade secrets.

How many storage accounts do you need?

A storage account represents **a collection of settings** like location, replication strategy, and subscription owner. You need *one storage account* for *every group of settings* that you want to apply to your data.

Do you have some data that is proprietary and some for public consumption? If so, you could enable virtual networks for the proprietary data and not for the public data. This will also require separate storage accounts. Various departments/budgets that should pay the bills?

A typical strategy is to start with an analysis of your data and create partitions that share characteristics like **location, billing, and replication strategy,** and then create **one storage account for each partition.**



Demo

Creating an Azure Storage Account using Azure Portal…

Knowledge Check

1. Suppose you have two video files stored as blobs. One of the videos is **business-critical** and requires a **replication policy** that creates multiple copies across geographically diverse datacenters.
   The other video is **non-critical**, and a **local replication** policy is sufficient.

   Which setup would you choose to satisfy both data diversity and cost sensitivity consideration?

3.0

Connect your App to Azure Storage - Blob Storage.

In Blob storage, every blob lives inside a *blob container*.
You can store an unlimited number of blobs in a container and an unlimited number of containers in a storage account.

Demo

Connecting our React-app to Azure Storage… displaying a list of blobs with virtual directories and defined MetaData.

Knowledge Check

1. How many access keys are provided for accessing your Azure storage account?

Answer:

4.0 Self Study

Azure Storage Security & Azure Defender: Advanced Threat Protection

Data breach - almost always due to
Cloud services not properly configured..

https://docs.microsoft.com/en-us/learn/modules/secure-azure-storage-account/

4.0
Self Study.
Data Storage more in-depth

Microsoft Learn - Store Data in Azure

https://docs.microsoft.com/en-us/learn/paths/store-data-in-azure/

Thank you.