

Breaking Data Communication Barriers

- 🌐 Global systems: different languages
- 🔧 Every platform: unique data dialect
- 💡 Universal solution: OData breakthrough
- ⚡ Result: seamless data connectivity





Current Digital Communication Chaos

 Mobile apps

JSON requirements

 Web dashboards

REST protocols

 Legacy ERP

XML standards

 IoT sensors

evolving languages daily

Web's Revolutionary Solution Model

01

📅 1990s chaos

similar communication problems

02

🌐 Universal rules

HTTP, HTML, URLs

03

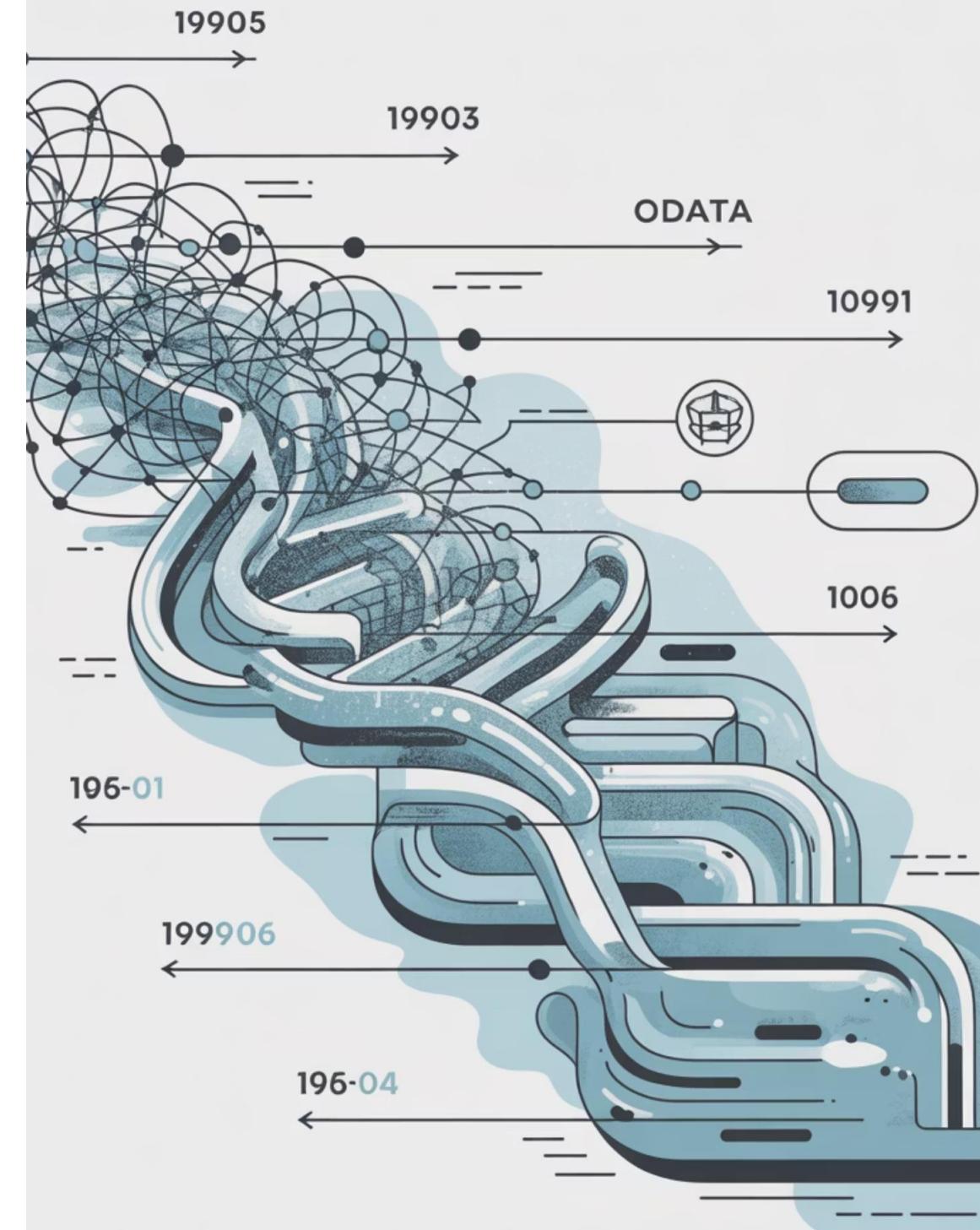
🔗 Breakthrough result

any browser, any server

04

✓ OData follows

same proven approach



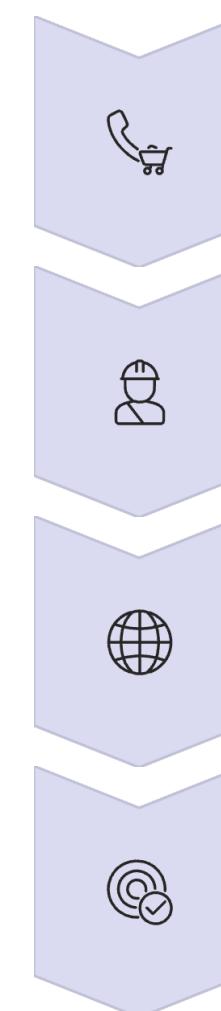
Netflix Connection Magic Explained



- 📱 Your phone: doesn't speak Netflix
- 🔌 Universal translator: HTTP connector
- 🎬 Netflix servers: same connector type
- ⚡ Simple request: "show me action movies"



OData Business Data Translation



Mobile app

OData connector

SAP system

Gateway OData connector

Universal interpreters

diplomatic translators

Same principle

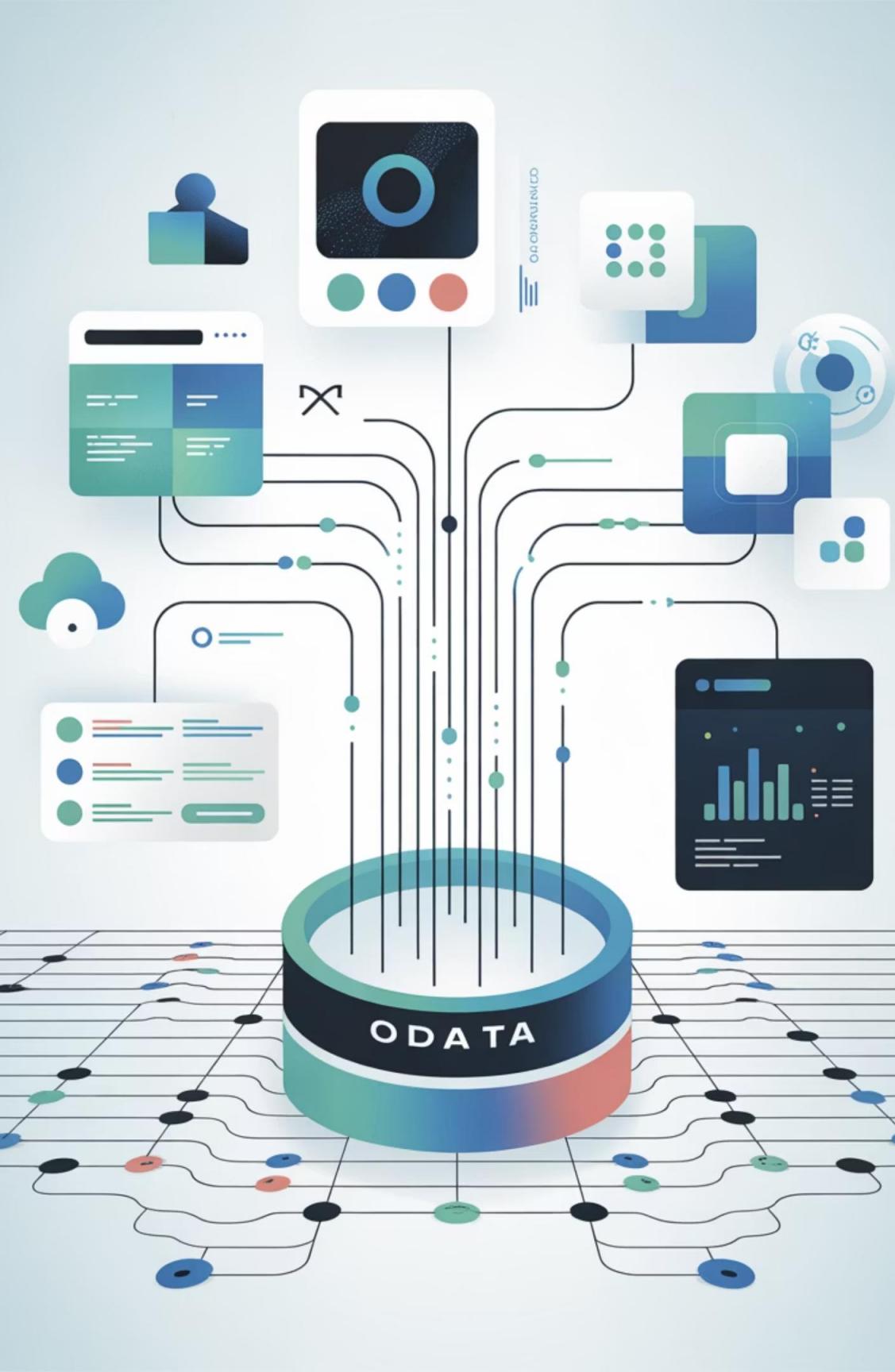
Netflix-style simplicity

Beyond Translation: Behavioral Contract

- ❑ OData promise:
standardized behavior
- ❑ HTTP parallel: browser
compatibility guarantee

- ❑ Universal connectivity:
any OData client
- ❑ Predictable interaction:
known protocols





Universal Client Compatibility



Fiori apps

native OData support



Power BI

direct OData connections



Custom mobile

standard integration



Excel

built-in OData capabilities

AFTER

Mindset Shift: From How to What

 Old question

"How do I connect?"

 New question

"What data needed?"

 OData guarantee

already connected

 Focus shift

connection to consumption



Real Impact: Maria's Transformation



Before

3 hours Monday data collection



After

15 minutes insight review



Solution

single Power BI report



Sources

five systems, one dashboard

Your OData Breakthrough Moment



Universal translator

any app, any data



Proven approach

World Wide Web model



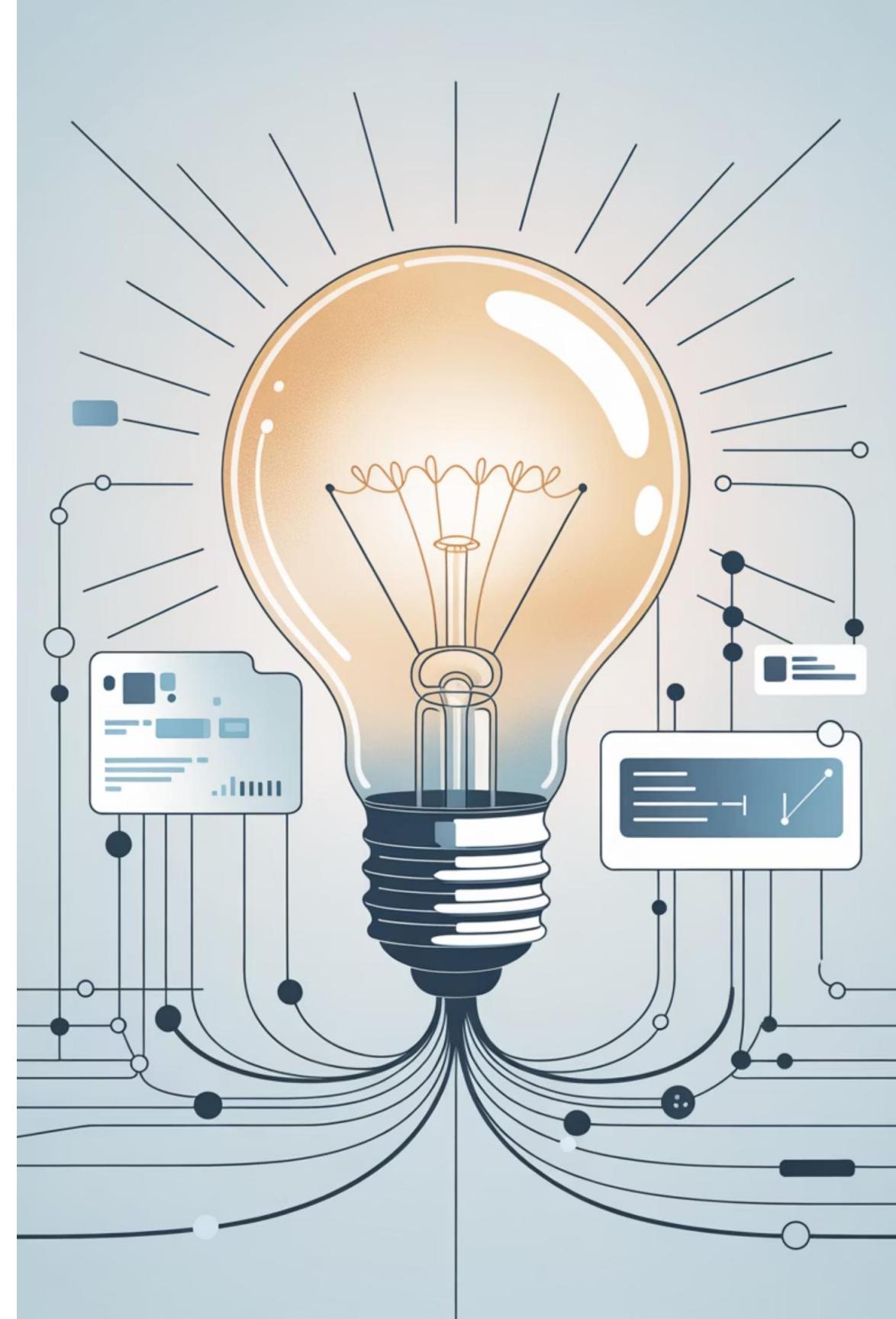
Common language

eliminates custom bridges



Focus shift

data needs over connections





Reflection Challenge

💡 Keep in mind: upcoming Gateway solution

🔍 Current pain points

manual data bridges

🔧 Custom connectors

unnecessary complexity

📊 Manual exports

system isolation

Next Lesson Preview

SAP Gateway: frustration to elegance

1

 Technology purpose
bridges not walls

2

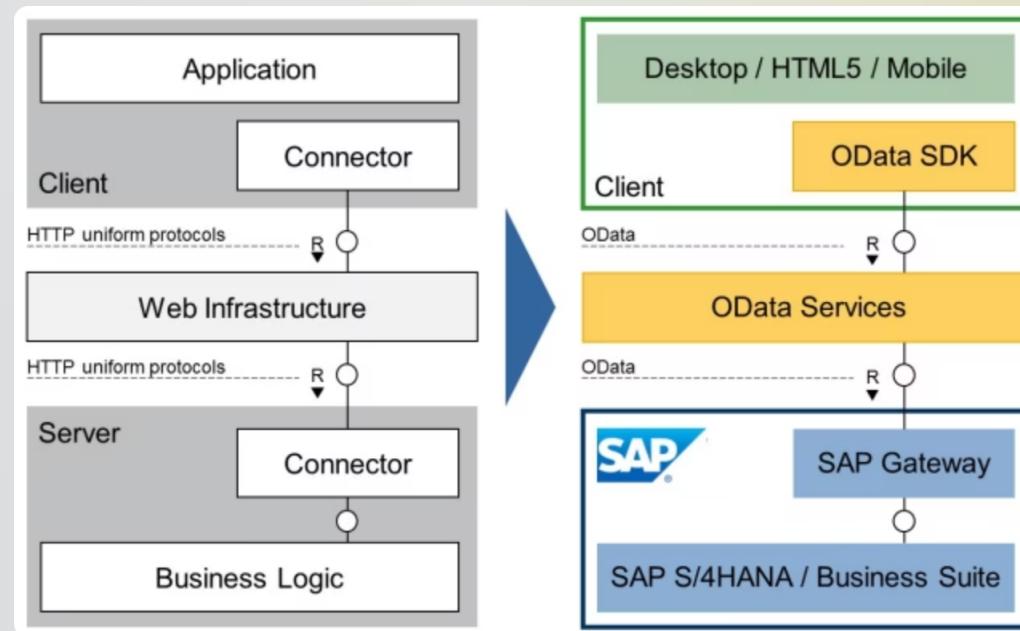
 Data to decisions
seamless flow

3

 Coming up
practical implementation

Because technology should create bridges, not walls

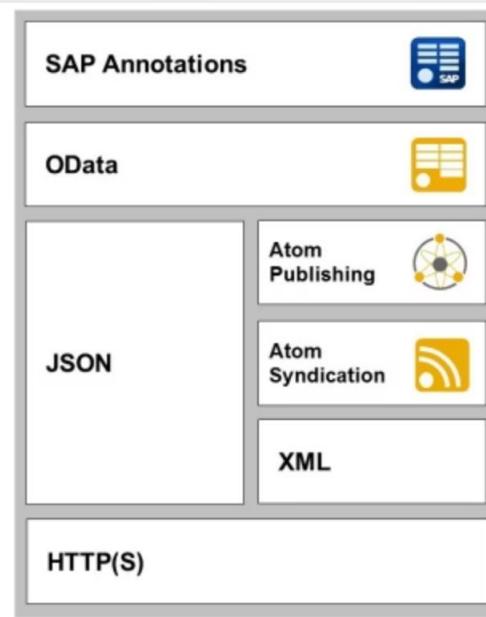
OData Foundation Technologies



- 🏗 Proven web standards: smart combination
- 🔌 ODBC for web: standardized transfer
- 🏡 House building analogy: established materials
- ⚡ Smart architecture: not reinventing basics

Core Technology Components

- SAP Annotations enhance OData by adding additional information like dictionary labels.
- OData defines the standard to transfer data using already established technologies. It is also called "ODBC for the web".
- JSON (Java Script Object Notation) defines the data format for the transferred data.
- XML in combination with Atom Publishing and Atom Syndication is the alternative to JSON with a wider support range.
- HTTP(S) is the network protocol for communication.



JSON format

lightweight, efficient transfer

XML/Atom

verbose, metadata-rich structure

HTTPS protocol

fundamental web communication

REST principles

web-friendly architecture

OData Standards Evolution



01

 Microsoft origin

now open standard

02

 OASIS management

industry governance

03

 REST foundation

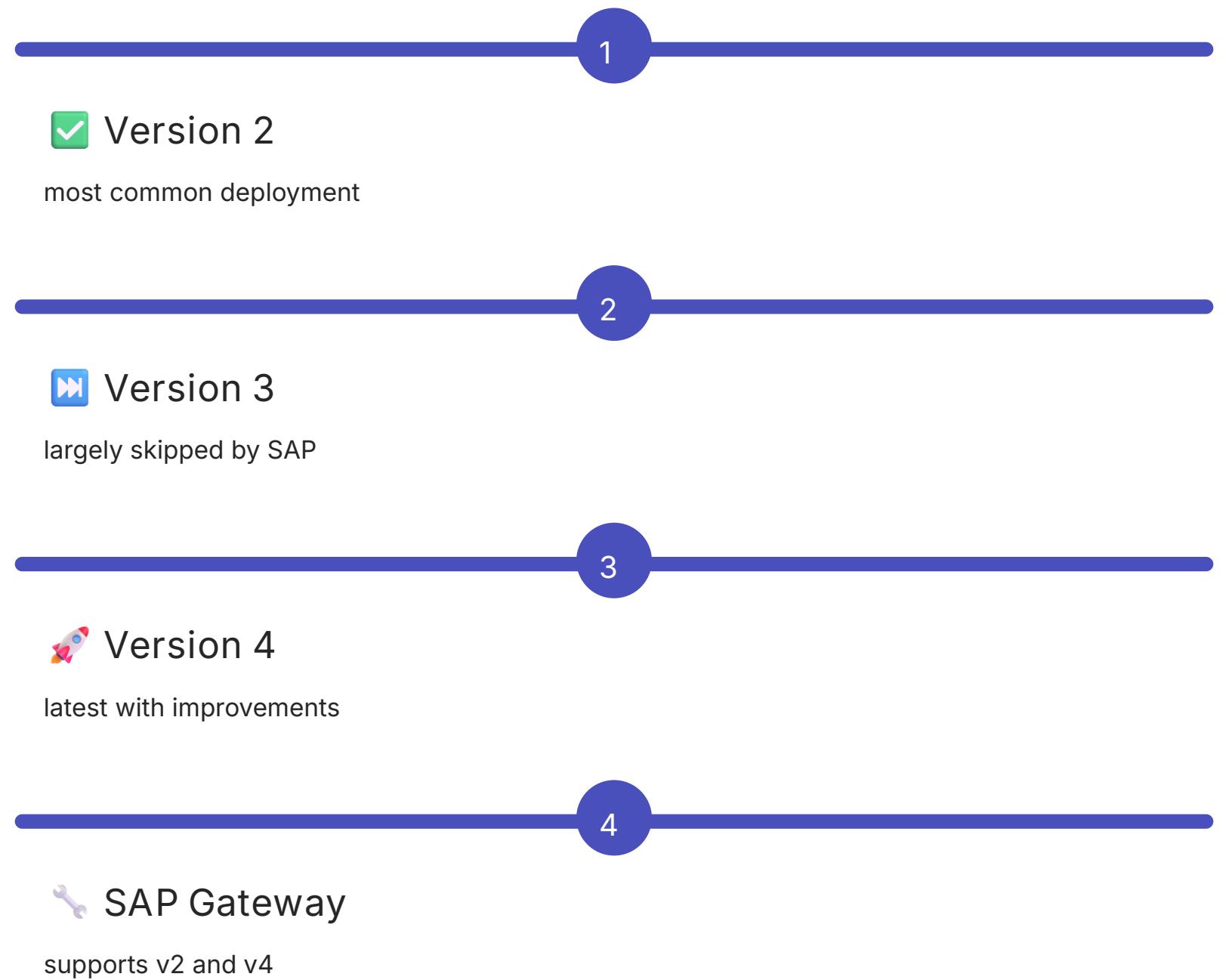
web-native approach

04

 Multiple versions

enterprise adoption

Version Landscape Overview



Format Trade-off Analysis

⚡ JSON benefits

efficient, smaller size

💻 JSON use case

modern applications

📊 XML benefits

rich metadata, detailed

🔧 XML use case

development scenarios

Real Performance Impact: David's Story

8s

⌚ XML
performance
load time

2s

⚡ JSON
improvement
load time

45KB

📊 Original size
XML format

12KB

📊 Reduced size
JSON format

📱 Challenge: Fiori app loading speed



Format Selection Strategy

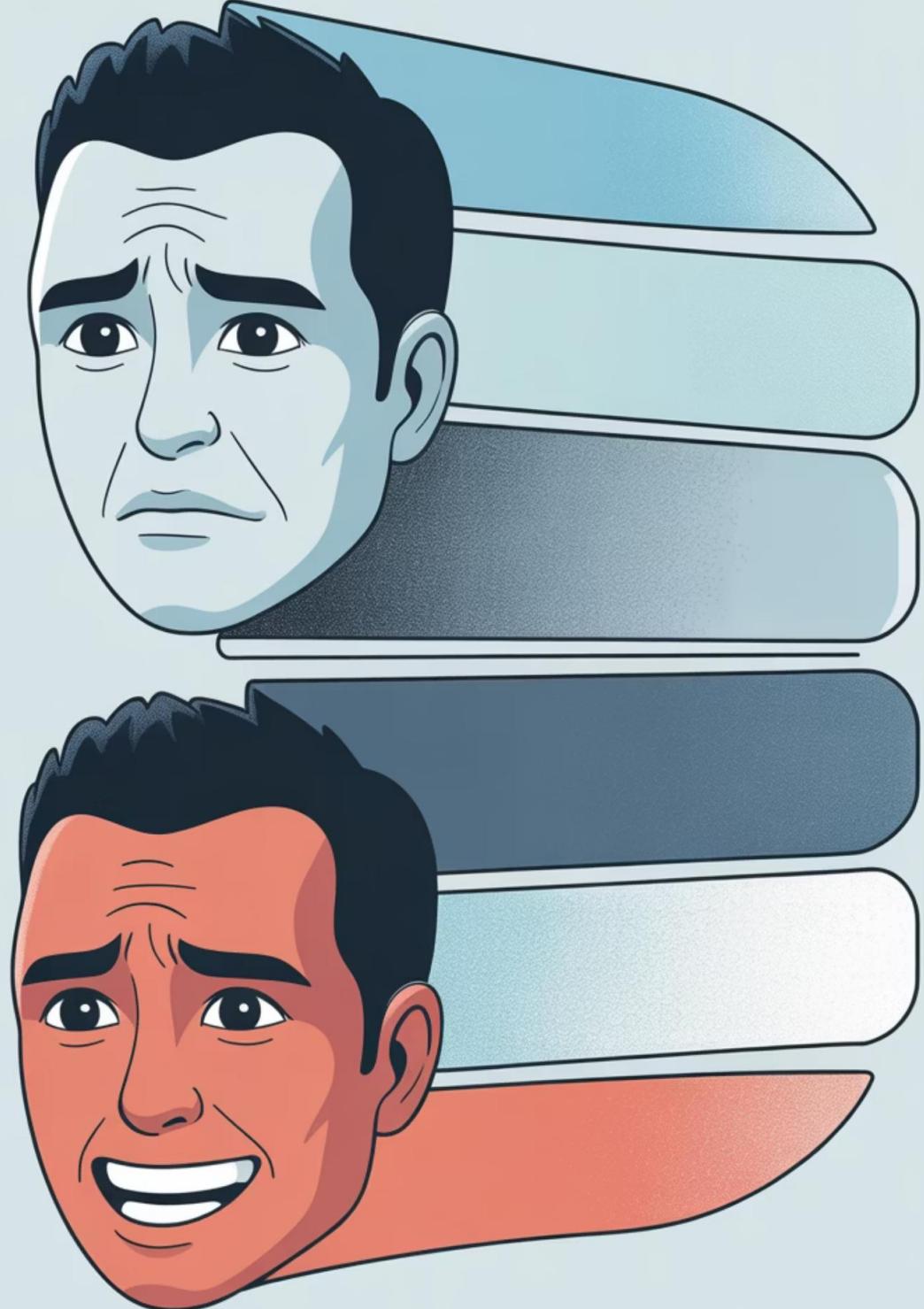
- 1  JSON advantage
lean, fast transfer
- 2  Bandwidth concern
critical consideration
- 3  Communication analogy
letter vs text
- 4  Resource efficiency
faster delivery

Atom

```
<entry xmlns="http://www.w3.org/2005/Atom">
  <id>http://schemas.microsoft.com/ado/2007/08/dataservices/metadata#ProductID=101153A053AA0.0000000</id>
  <updated>2015-09-10T15:03:48.0000000Z</updated>
  <category term="IWEBP/GMSAMPLE_BASIC.Product" scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"></category>
  <link href="ProductSet('HT-1000')/ToSupplier" rel="edit" title="ToSupplier"/>
  <link href="ProductSet('HT-1000')/ToSalesOrderLineItems" rel="edit" title="ToSalesOrderLineItems"/>
  <link href="ProductSet('HT-1000')/ToSalesOrderLineItems" type="application/atom+xml&#034;entry&#034;" title="ToSalesOrderLineItems"/>
  <link href="ProductSet('HT-1000')/ToSupplier" type="application/atom+xml&#034;entry&#034;" title="ToSupplier"/>
  <content type="application/xml">
    <atom:properties>
      <d:ProductID>HT-1000</d:ProductID>
      <d:TypeCode>PR</d:TypeCode>
      <d:Category>Notebooks</d:Category>
      <d:Name>Notebook Basic 15</d:Name>
      <d:NameLanguage>EN</d:NameLanguage>
      <d:Description>Notebook Basic 15 with 2,80 GHz quad core, 15" LCD, 4 GB DDR3 RAM, 500 GB Hard Disc, Windows 8 Pro</d:Description>
      <d:DescriptionLanguage>EN</d:DescriptionLanguage>
      <d:SupplierID>1000000000</d:SupplierID>
      <d:SupplierName>SAP</d:SupplierName>
      <d:TaxTariffCode>1</d:TaxTariffCode>
      <d:MeasureUnit>Kg</d:MeasureUnit>
      <d:WeightMeasure>4.200</d:WeightMeasure>
      <d:WeightUnit>KG</d:WeightUnit>
      <d:CurrencyCode>EUR</d:CurrencyCode>
      <d:Price>896.00</d:Price>
      <d:Width>0.30</d:Width>
      <d:Depth>0.18</d:Depth>
      <d:Height>0.03</d:Height>
      <d:CreatedAt>2015-09-10T15:03:48.0000000Z</d:CreatedAt>
      <d:ChangedAt>2015-09-10T15:03:48.0000000Z</d:ChangedAt>
    </atom:properties>
  </content>
</entry>
```

JSON

```
- 4: {
  + _metadata: {..},
  ProductID: "HT-1000",
  TypeCode: "PR",
  Category: "Notebooks",
  Name: "Notebook Basic 15",
  NameLanguage: "EN",
  Description: "Notebook Basic 15 with 2,80 GHz quad core, 15\" LCD, 4 GB DDR3 RAM, 500 GB Hard Disc, Windows 8 Pro",
  DescriptionLanguage: "EN",
  SupplierID: "1000000000",
  SupplierName: "SAP",
  TaxTariffCode: "1",
  MeasureUnit: "KG",
  WeightMeasure: "4.200",
  WeightUnit: "KG",
  CurrencyCode: "EUR",
  Price: "896.00",
  Width: "0.30",
  Depth: "0.18",
  Height: "0.03",
  CreatedAt: "2015-09-10T15:03:48.0000000Z",
  ChangedAt: "2015-09-10T15:03:48.0000000Z",
  + ToSalesOrderLineItems: {..},
  + ToSupplier: {..}
}
```



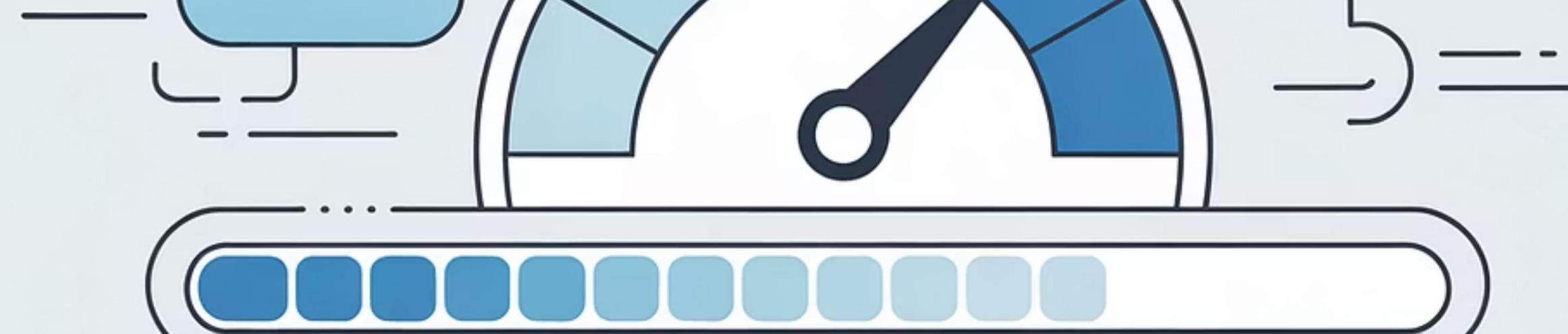
Performance Impact Assessment

Current Issues

- 🚨 Current issues: slow loading times
- 📱 Mobile complaints: app performance

The Difference

- 🎯 Format choice: user experience difference
- ⚡ Outcome: frustration vs delight



Trust Through Speed

⌚ Every millisecond: user experience impact



🤝 Fast loading

builds user trust



😱 Slow performance

tests patience



🌐 Connected world

speed expectations

Daily Data Interactions

-  Create: new notes, posts
-  Read: articles, information
-  Update: social profiles, content
-  Delete: old emails, files

CRUD Operations Foundation

Operation on resource	HTTP verb
Create	POST
Read	GET
Update	PUT
Delete	DELETE

Create

fundamental data action

Read

information retrieval

Update

modify existing data

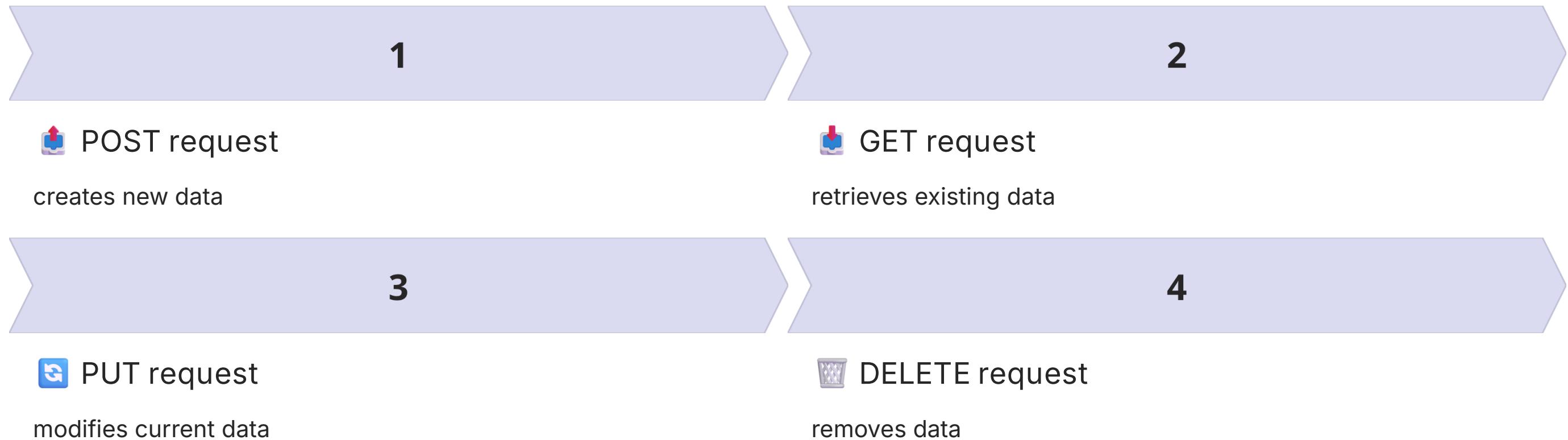
Delete

remove data/resources

OData integration: web-native approach

These are fundamental actions, often summarized as CRUD

HTTP Verb Mapping



Revolutionary Simplification

-  No complex database commands
-  Standard web requests: browser-familiar
-  Addressable resources: unique URLs
-  Web-native data interaction

This means something revolutionary

Social Media Parallel

👍 Like button

POST request

👤 Profile view

GET request

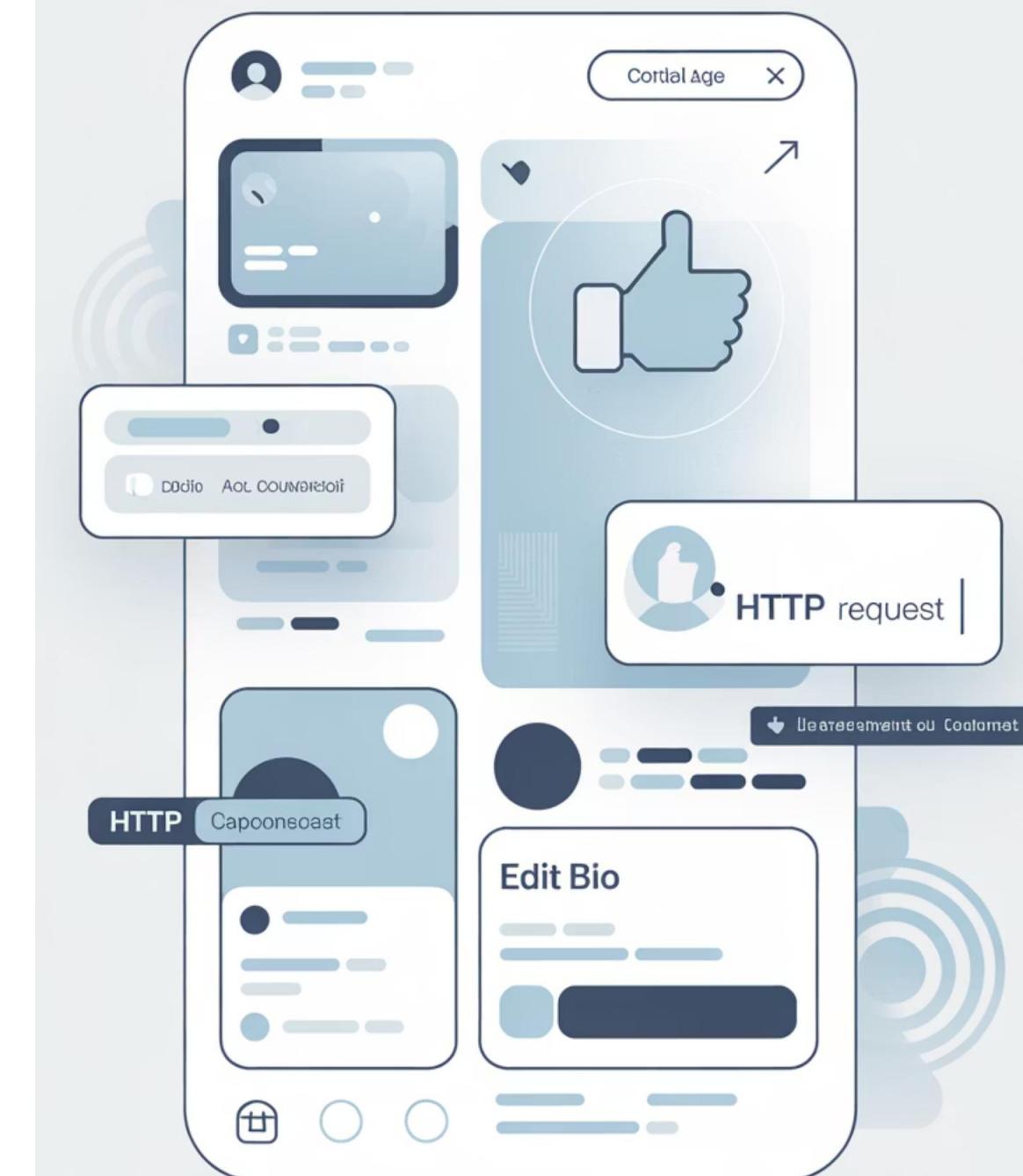
📝 Bio edit

PUT request

🏢 OData business

same principles

When you click like on a social media post



Real Impact: Jennifer's Transformation

Before

-  Challenge: mobile customer updates
-  Before: complex database connections

After

-  After: simple PUT requests
-  Result: automatic SAP updates

Universal Language Advantage



Browsers

native HTTP support

Dashboards

fluent communication



Mobile apps

built-in capability



Business systems

same language

Your web browser already knows how to send

Intuitive Data Management



 Database actions

HTTP requests



 Phone photos

same logic



 Business data

familiar actions



 Universal web language

Current Challenge Assessment

 Multiple interfaces

team confusion

 Web familiarity

existing knowledge

 Training time

adoption barriers

 Effortless adoption

natural transition

Think about your current data challenges



Technology Speaks Human Language



🗣 Familiar language
easier adoption



⚡ Effortless learning
natural transition



🏃 Painful training
eliminated barriers



🤝 User understanding
immediate comfort

Because when technology speaks the language people already understand

OData Testing Fundamentals

- 💡 Testing equals building importance
- 📊 Query skills: essential development
- 🎯 Pre-built setup: focus learning
- ⚡ CDS View: Z_PRODUCTS_ODATA ready

Service Architecture Overview

01

CDS View: Z_PRODUCTS_ODATA

 Foundation layer for data access

02

Service Description: Z_PRODUCTS_SD

 Service definition and metadata

03

Service Binding: Z_PRODUCTS_SB

 Connection to OData protocol

 Inheritance over new builds



Service Maintenance Check

🔧 Transaction: /IWEND/MAINT_SERVICE

- Service registration status
- Technical service details
- First troubleshooting stop

Gateway Client Power Tool

⚡ Transaction:
/IWFND/GW_CLIENT
Built-in SAP testing interface

🔧 Technical Details Visible
Complete request and response inspection

💪 Favorite Testing Method
Most reliable way to validate services

Service Document Foundation

URI: /sap/opu/odata/sap/Z_PRODUCTS_SB/

 HTTP 200: success response

Confirms service is accessible and responding correctly

 EntitySets: table of contents

Lists all available data collections in the service

 Products EntitySet target
Primary collection for querying product data

Metadata Blueprint Analysis

 URI: /\$metadata endpoint

-  Service complete blueprint
-  EntityType: Z_PRODUCTS_ODATATYPE
-  Pre-query essential check

Flight Data Structure



AirlineID

Three-character code identifying the airline carrier



ConnectionID

Four-character flight number for the route



FlightDate

Operation schedule date for the flight



Currency

Pricing code for financial transactions

Aircraft and Capacity Fields



 **PlaneType:** aircraft model

Identifies the specific aircraft type used for the flight



 **MaxSeats:** total capacity

Maximum number of passengers the aircraft can accommodate



 **OccupiedSeats:** passenger count

Current number of booked seats on the flight



 **EntitySet:** Z_PRODUCTS_ODATA

The collection containing all flight data records

Basic Data Querying

 URI: /Z_PRODUCTS_ODATA EntitySet

 "Give everything" query

Retrieves all records from the EntitySet without any filters or limitations

 Production volume consideration

Be cautious with large datasets in live environments

 Testing perfect approach

Ideal for development and testing scenarios to see complete data structure

Airline Filtering Examples



 Singapore Airlines

`AirlineID eq 'SQ'`



 United Airlines

`AirlineID eq 'UA'`

 Standardized OData syntax

 Universal service consistency

Capacity and Currency Filters



High occupancy

```
OccupiedSeats gt 300
```

Flights with over 300 passengers

US operations

```
Currency eq 'USD'
```

Flights priced in US dollars

Business logic implementation

Apply real-world filtering criteria to narrow down results

Real-world query patterns

Common scenarios used in production applications

OData Filter Operators



`eq`: equals comparison

Matches exact values



`ne`: not equals

Excludes specific values



`gt/lt`: greater/less than

Strict comparisons



`ge/le`: greater/less equal

Inclusive comparisons

Performance Optimization Techniques

\$top: sample size control

Limit the number of records returned for quick testing

Performance impact crucial

These parameters significantly affect response times

\$select: specific fields only

Request only the columns you need to reduce payload

Complex combination queries

Combine multiple parameters for optimal results

Advanced Query Patterns



Multiple airline filtering

Filter for United OR American Airlines using logical operators



Aircraft type analysis

Query specific plane models for operational insights



Wide-body operations

Identify large aircraft for capacity planning



European flights sorting

Order results by date, occupancy, or other criteria

JSON vs XML Formats

XML: default format

Standard OData response format, verbose but structured

JSON: modern preference

 `$format=json` parameter

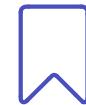
 Fiori application consumption

Browser Testing Benefits



🌐 Direct browser access

Test OData services directly in Chrome, Firefox, or any modern browser without additional tools



🔖 Bookmark useful queries

Save frequently used query URLs for quick access during development



👥 Team URL sharing

Easily share specific queries with colleagues for collaboration



⚡ Quick response checking

Instantly verify service responses without complex setup

Common Troubleshooting Issues



✖ HTTP 404: service missing

Service not registered or URL incorrect - check service binding and activation



✳ HTTP 500: internal errors

Server-side issues in CDS view or service implementation - review error logs



🔍 Empty results: filter issues

Filter syntax errors or no matching data - verify filter operators and values



🔒 Authentication session expiry

Session timeout requiring re-authentication - refresh credentials

Performance Monitoring Strategy

- ⏳ Response time tracking

Monitor how long queries take to return results

- 📈 Testing vs production volumes

Understand the difference between small test datasets and large production data

- 🎯 \$top, \$select, \$filter usage

Apply optimization parameters to improve query performance

- 💡 Future self consideration

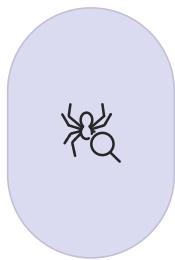
Design queries that will scale as data grows over time

Real-World Query Patterns



🎯 Specific flight connection

Query exact flight by airline and connection ID for detailed information



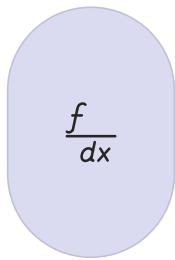
🔍 Aircraft type searching

Find all flights using a particular plane model for fleet analysis



📊 Occupancy ordering

Sort flights by passenger count to identify high-demand routes



🔢 Count without data retrieval

Use \$count to get totals without fetching full records

Service Validation Checklist

Service document loads

Verify the base service URL returns HTTP 200 and shows EntitySets

Basic entity query

Test retrieving data from the main EntitySet without filters

Metadata completeness check

Confirm /\$metadata endpoint displays all EntityTypes and properties

Filtering functionality test

Apply sample filters to ensure query parameters work correctly

Consumer-Focused Development

End-user experience focus

Always consider how the service will be consumed by real users

Fiori application consumption

This is exactly how Fiori applications will interact with your service

Performance characteristic observation

Pay attention to response times and data volumes during testing

Consumer mindset adoption

Think like the application developer who will use your OData service

OData Testing Fundamentals

 Testing equals building importance

 Query skills: essential development

 Pre-built setup: focus learning

 Real-world inheritance patterns

Service Structure Overview

01

 CDS View: Z_PRODUCTS_ODATA

03

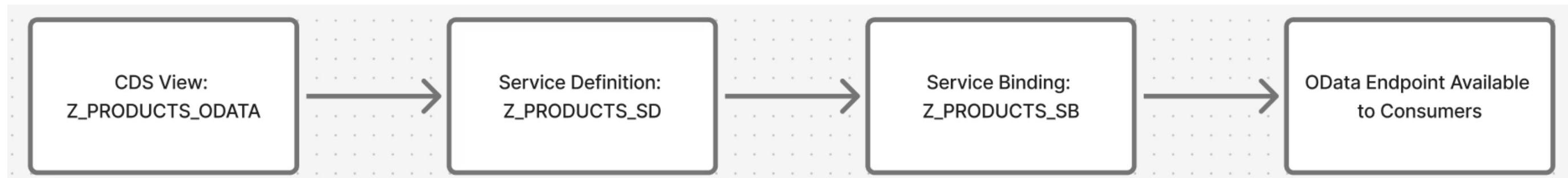
 Service Binding: Z_PRODUCTS_SB

02

 Service Description: Z_PRODUCTS_SD

04

 Inheritance over new builds



Service Maintenance Check

The screenshot shows the SAP Activate and Maintain Services interface. The top navigation bar includes 'Activate and Maintain Services' and various icons for search, refresh, and system status. Below the header, there are two main sections: 'Service Catalog' and 'ICF Nodes'.
Service Catalog: This section displays a list of technical services with columns for Type, Technical Service Name, Service Description, External Service Name, Neg., OAut., Soft State, and Processing Mode. Some entries are marked as 'Not Supported' or 'Co-deployed only'.
ICF Nodes: This section shows a table of ICF nodes with columns for Status, ICF Node, Session Time-out, Soft State, and Description. One node, 'ZUM_FLIGHT_R_V2', is highlighted with a red border.
At the bottom of the interface, there are tabs for 'ICF Node', 'Call Browser', 'SAP Gateway Client', and a status bar indicating 'A4H (1) 001 vhc4h1 INS'.

🔍 Transaction:
/IWFND/MAINT_SERVICE

✓ Service registration status

📝 Technical service details

🎯 First troubleshooting stop

Gateway Client Power Tool

 Transaction: /IWFND/GW_CLIENT

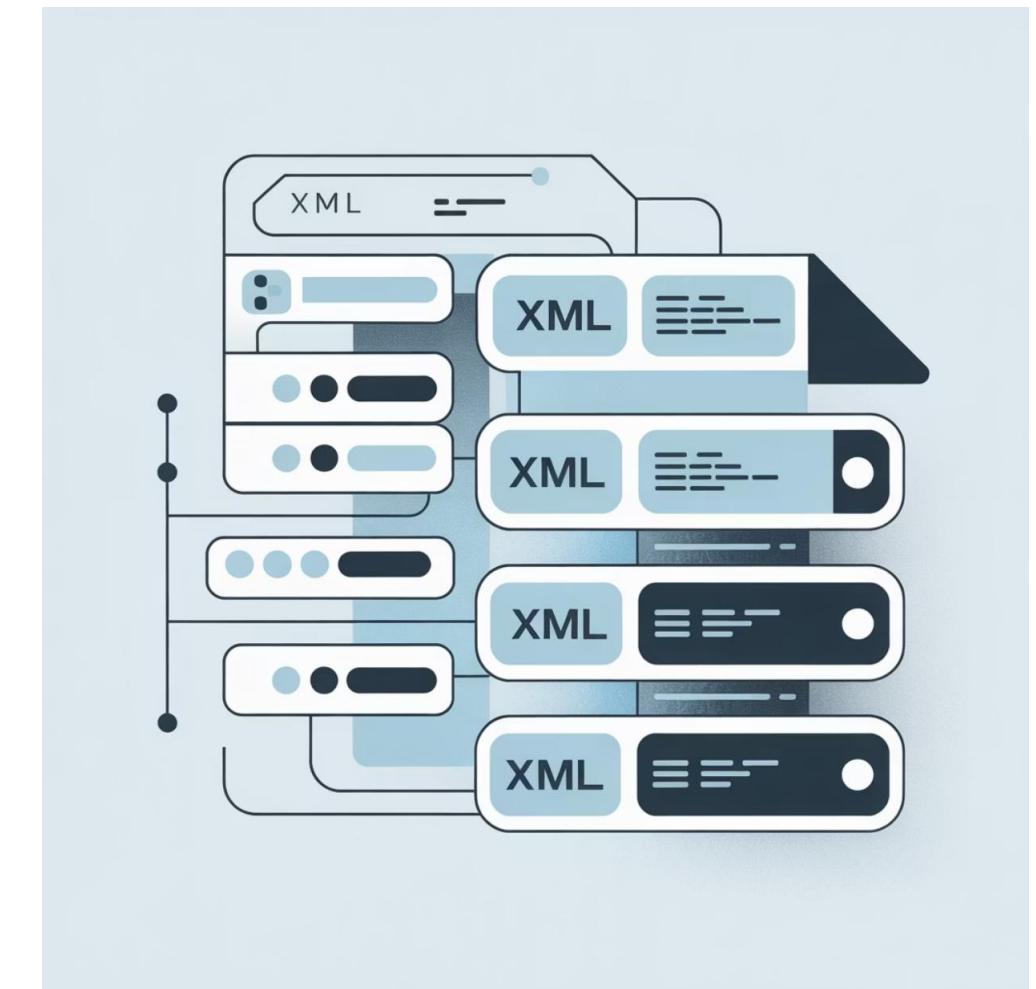
 Built-in SAP testing

 Technical details visible

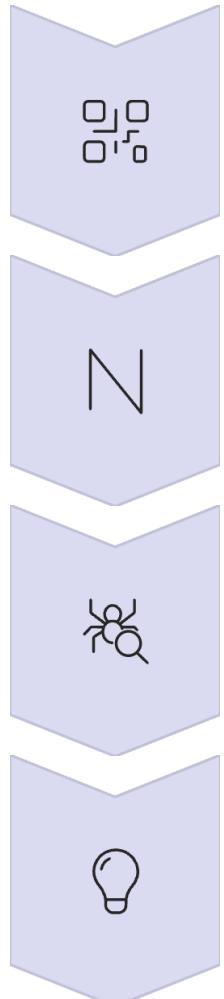
 Favorite testing method

Service Document Foundation

-   URI: /sap/opu/odata/sap/Z_PRODUCTS_SB/
-   HTTP 200: success response
-   EntitySets: table of contents
-   Products EntitySet target



Metadata Blueprint Analysis



- 💡 URI: /\$metadata endpoint
- 📐 Service complete blueprint
- 🔍 EntityType: field definitions
- 💡 Pre-query essential check



Basic Data Querying

URI: /Products EntitySet

🎯 "Give everything" query

⚠ Production volume consideration

✓ Testing perfect approach



OData Filtering Power



🔍 \$filter: standardized syntax



📝 ProductType eq 'FINISHED'



💰 Price gt 100



🌐 Universal OData consistency

Common Filter Operators

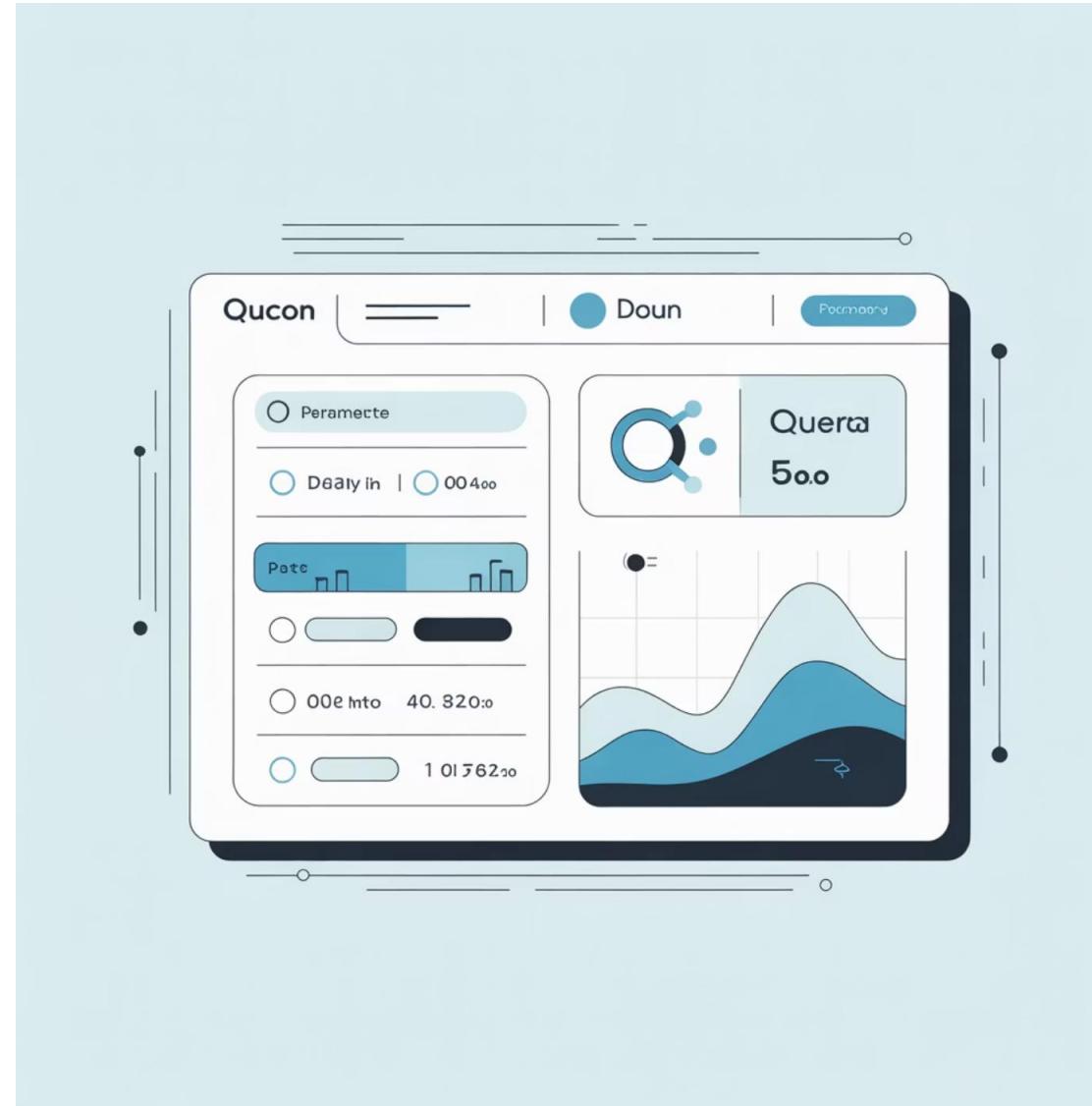
  eq: equals comparison

  ne: not equals

  gt/lt: greater/less than

  ge/le: greater/less equal

Data Limiting Techniques



1 2
3 4 \$top: sample size control

\$select: specific fields

⚡ Performance optimization crucial

⌚ Combination queries possible

JSON vs XML Formats



XML: default format



JSON: modern preference



\$format=json parameter



✨ Cleaner application consumption

Browser Testing Simplicity

 Direct browser access

 Bookmark useful queries

 Team URL sharing

 Quick response checking



Common Troubleshooting Issues



✗ HTTP 404: service missing



✳️ HTTP 500: internal errors



🔍 Empty results: filter issues



🔒 Authentication session expiry

Performance Considerations

⌚ Response time monitoring

📈 Testing vs production volumes

🎯 \$top, \$select, \$filter usage

💡 Future self appreciation



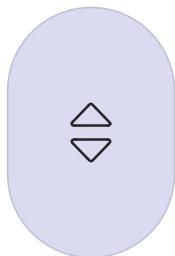
Real-World Query Patterns



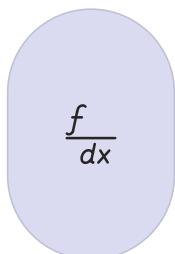
 Specific product retrieval



 Partial name searching



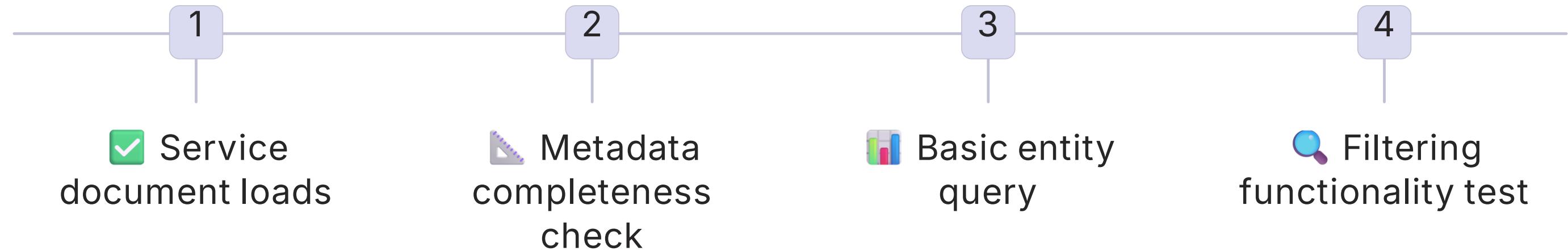
 Result ordering capabilities



 Count without data



Service Validation Checklist



Consumer-Focused Development



- End-user experience focus
- Fiori application consumption
- Performance characteristic observation
- Consumer mindset adoption



Data Discovery Challenge

App question: what data available?

📚 Library analogy: catalog needed

📋 Clear definitions: accessible information

🔍 Discovery solution: structured approach

Service Document Foundation

Main URL access

service document

Entity sets

similar data collections

Examples

Products, Customers

Available resources

clear listing

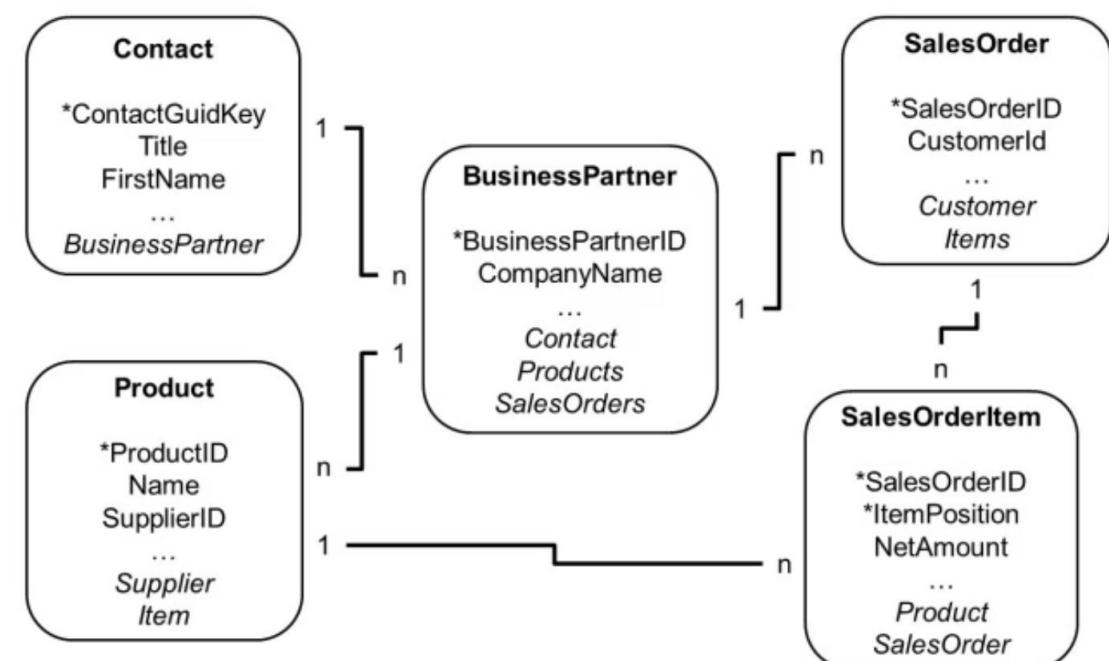
To consume an OData service for read, you just need a browser and the OData service URI. This leads to the service document. To get data from the service, add the name of an entity of the service to the base URI. You get a list of entities of that type, which could be the content of a database table.

```
<app:service xmlns:app="http://www.w3.org/2007/app" xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
  xmlns:sap="http://www.sap.com/Protocols/SAPData" xml:lang="de"
  xmlns:base="http://
    /sap/opu/odata/IWBEP/GWSAMPLE_BASIC/">
  <app:workspace>
    <atom:title type="text">Data</atom:title>
    <app:collection sap:content-version="1" href="BusinessPartnerSet">
      <atom:title type="text">BusinessPartnerSet</atom:title>
      <sap:member-title>BusinessPartner</sap:member-title>
    </app:collection>
    <app:collection sap:content-version="1" href="ProductSet">
      <atom:title type="text">ProductSet</atom:title>
      <sap:member-title>Product</sap:member-title>
    </app:collection>
    <app:collection sap:updatable="false" sap:content-version="1" href="SalesOrderSet">
      <atom:title type="text">SalesOrderSet</atom:title>
      <sap:member-title>SalesOrder</sap:member-title>
    </app:collection>
    <app:collection sap:content-version="1" href="SalesOrderLineItemSet">
      <atom:title type="text">SalesOrderLineItemSet</atom:title>
      <sap:member-title>SalesOrderLineItem</sap:member-title>
    </app:collection>
    <app:collection sap:content-version="1" href="ContactSet">
      <atom:title type="text">ContactSet</atom:title>
      <sap:member-title>Contact</sap:member-title>
    </app:collection>
  </app:workspace>
  <atom:link rel="self"
    href="http://
      /sap/opu/odata/IWBEP/GWSAMPLE_BASIC/"/>
  <atom:link rel="latest-version"
    href="http://
      /sap/opu/odata/IWBEP/GWSAMPLE_BASIC/"/>
</app:service>
```

Entity Data Model Blueprint

E EDM structure: data templates

- Entity types: Product example
- Properties: ID, Name, Price
- Relationships: Product to Supplier



Navigation Properties Power



Runtime connections
entity jumping

Related data
seamless access



Data relationships
clear mapping

Dynamic navigation
real-time linking

Metadata Document Access



URL addition: \$metadata



Machine-readable: comprehensive description



Complete blueprint: types, properties, relationships



Full structure: instant access

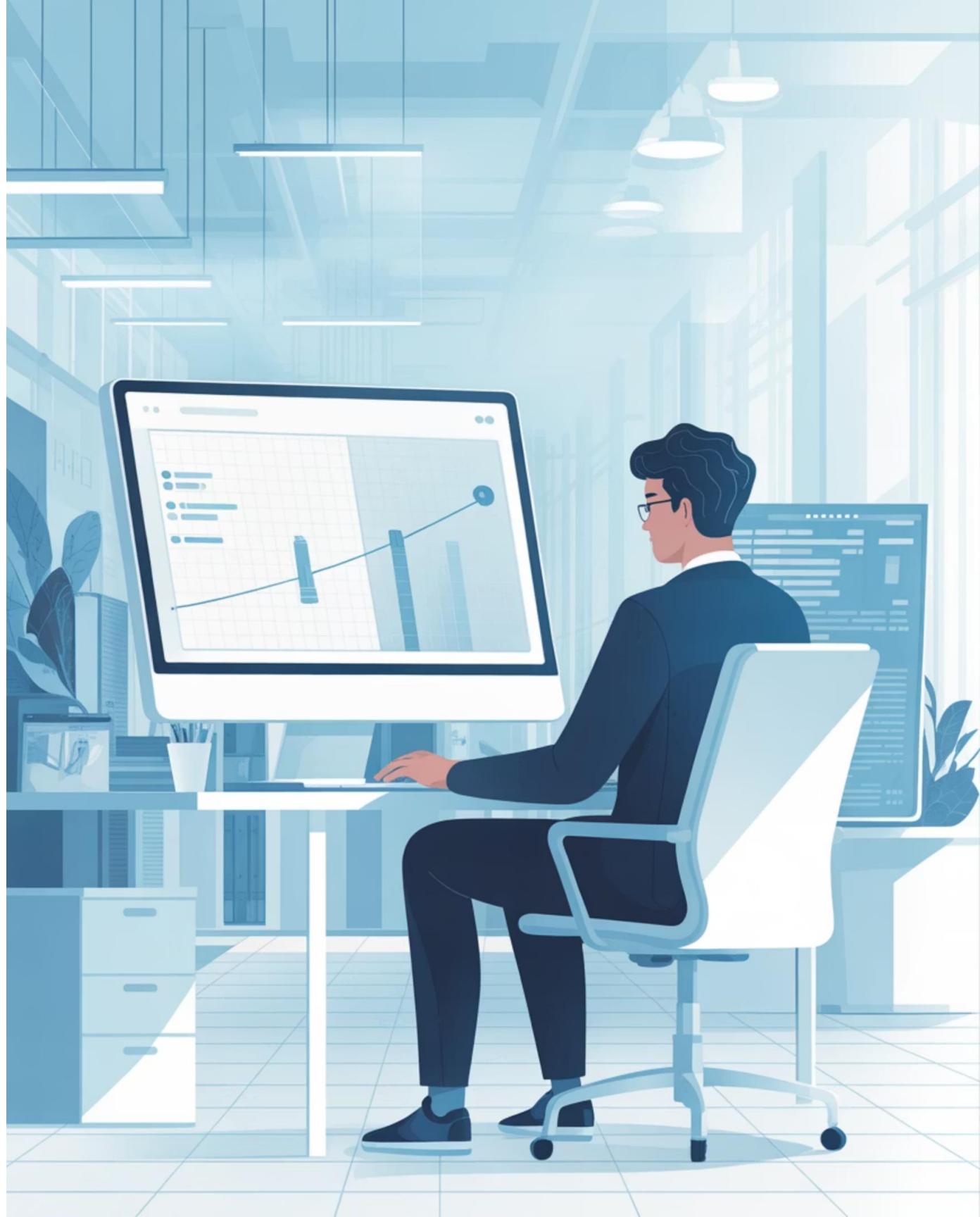
```
<edmx:Edmx xmlns:edmx="http://schemas.microsoft.com/ado/2007/06/edmx" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
  xmlns:sap="http://www.sap.com/Protocols/SAPData" Version="1.0">
  <edmx:DataServices m:DataServiceVersion="2.0">
    <Schema xmlns="http://schemas.microsoft.com/ado/2008/09/edm" Namespace="/IWBEP/GWSAMPLE_BASIC" xml:lang="de" sap:schema-version="1">
      <EntityType Name="BusinessPartner" sap:content-version="1">...</EntityType>
      <EntityType Name="Product" sap:content-version="1">
        <Key>
          <PropertyRef Name="ProductID"/>
        </Key>
        <Property Name="ProductID" Type="Edm.String" Nullable="false" MaxLength="10" sap:label="Produkt-ID" sap:updatable="false"/>
        <Property Name="TypeCode" Type="Edm.String" Nullable="false" MaxLength="2" sap:label="Typcode"/>
        <Property Name="Category" Type="Edm.String" Nullable="false" MaxLength="40" sap:label="Kategorie"/>
        <Property Name="Name" Type="Edm.String" Nullable="false" MaxLength="255" sap:sortable="false" sap:filterable="false"/>
        <Property Name="NameLanguage" Type="Edm.String" MaxLength="2" sap:label="Sprache" sap:createable="false" sap:updatable="false" sap:sortable="false" sap:filterable="false"/>
        <Property Name="Description" Type="Edm.String" MaxLength="255" sap:sortable="false" sap:filterable="false"/>
        <Property Name="DescriptionLanguage" Type="Edm.String" MaxLength="2" sap:label="Sprache" sap:createable="false" sap:updatable="false" sap:sortable="false" sap:filterable="false"/>
        <Property Name="SupplierID" Type="Edm.String" Nullable="false" MaxLength="10" sap:label="Geschäftspartner-ID"/>
        <Property Name="SupplierName" Type="Edm.String" MaxLength="80" sap:label="Firma" sap:createable="false" sap:updatable="false"/>
        <Property Name="TaxTarifCode" Type="Edm.Byte" Nullable="false" sap:label="Steuertarifcode"/>
        <Property Name="MeasureUnit" Type="Edm.String" Nullable="false" MaxLength="3" sap:label="Maßeinheit" sap:semantics="unit-of-measure"/>
        <Property Name="WeightMeasure" Type="Edm.Decimal" Precision="13" Scale="3" sap:unit="WeightUnit" sap:label="Gewicht"/>
        <Property Name="WeightUnit" Type="Edm.String" MaxLength="3" sap:label="Maßeinheit" sap:semantics="unit-of-measure"/>
        <Property Name="CurrencyCode" Type="Edm.String" Nullable="false" MaxLength="5" sap:label="Währungscode" sap:semantics="currency-code"/>
        <Property Name="Price" Type="Edm.Decimal" Precision="16" Scale="3" sap:unit="CurrencyCode" sap:label="Preis"/>
        <Property Name="Width" Type="Edm.Decimal" Precision="13" Scale="3" sap:unit="DimUnit" sap:label="Maßangaben"/>
        <Property Name="Depth" Type="Edm.Decimal" Precision="13" Scale="3" sap:unit="DimUnit" sap:label="Maßangaben"/>
        <Property Name="Height" Type="Edm.Decimal" Precision="13" Scale="3" sap:unit="DimUnit" sap:label="Maßangaben"/>
        <Property Name="DimUnit" Type="Edm.String" MaxLength="3" sap:label="Maßeinheit" sap:semantics="unit-of-measure"/>
        <Property Name="CreatedAt" Type="Edm.DateTime" Precision="7" sap:label="Zeitstempel" sap:createable="false" sap:updatable="false"/>
        <Property Name="ChangedAt" Type="Edm.DateTime" Precision="7" ConcurrencyMode="Fixed" sap:label="Zeitstempel" sap:createable="false" sap:updatable="false"/>
        <NavigationProperty Name="ToSalesOrderLineItems" Relationship="/IWBEP/GWSAMPLE_BASIC.Assoc_Product_SalesOrderLineItems"
          FromRole="FromRole_Assoc_Product_SalesOrderLineItems" ToRole="ToRole_Assoc_Product_SalesOrderLineItems"/>
        <NavigationProperty Name="ToSupplier" Relationship="/IWBEP/GWSAMPLE_BASIC.Assoc_BusinessPartner_Products"
          FromRole="ToRole_Assoc_BusinessPartner_Products" ToRole="FromRole_Assoc_BusinessPartner_Products"/>
      </EntityType>
      <EntityType Name="SalesOrder" sap:content-version="1">...</EntityType>
      <EntityType Name="SalesOrderLineItem" sap:content-version="1">...</EntityType>
      <EntityType Name="Contact" sap:content-version="1">...</EntityType>
    ...
  </Schema>
</edmx:DataServices>
</edmx:Edmx>
```

By adding the OData option \$metadata to the service root URI, the metadata document of the service is shown. The whole EDM is defined here and available at runtime. Application developers and all the wizards in development environments create their applications based on this information.

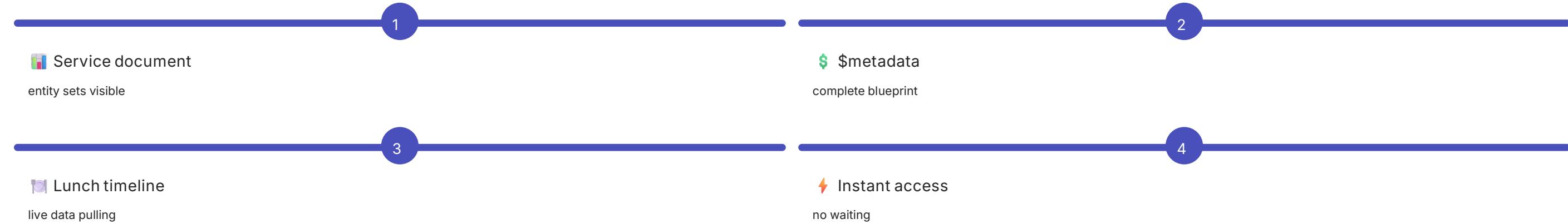
Real Impact: Marcus's Success

- Challenge: customer orders mobile app
- No calls: database administrator
- Simple access: service URL browser
- 15 minutes: complete understanding

[https://services.odata.org/V4/Northwind/Northwind.svc/\\$metadata](https://services.odata.org/V4/Northwind/Northwind.svc/$metadata)



Self-Service Data Discovery



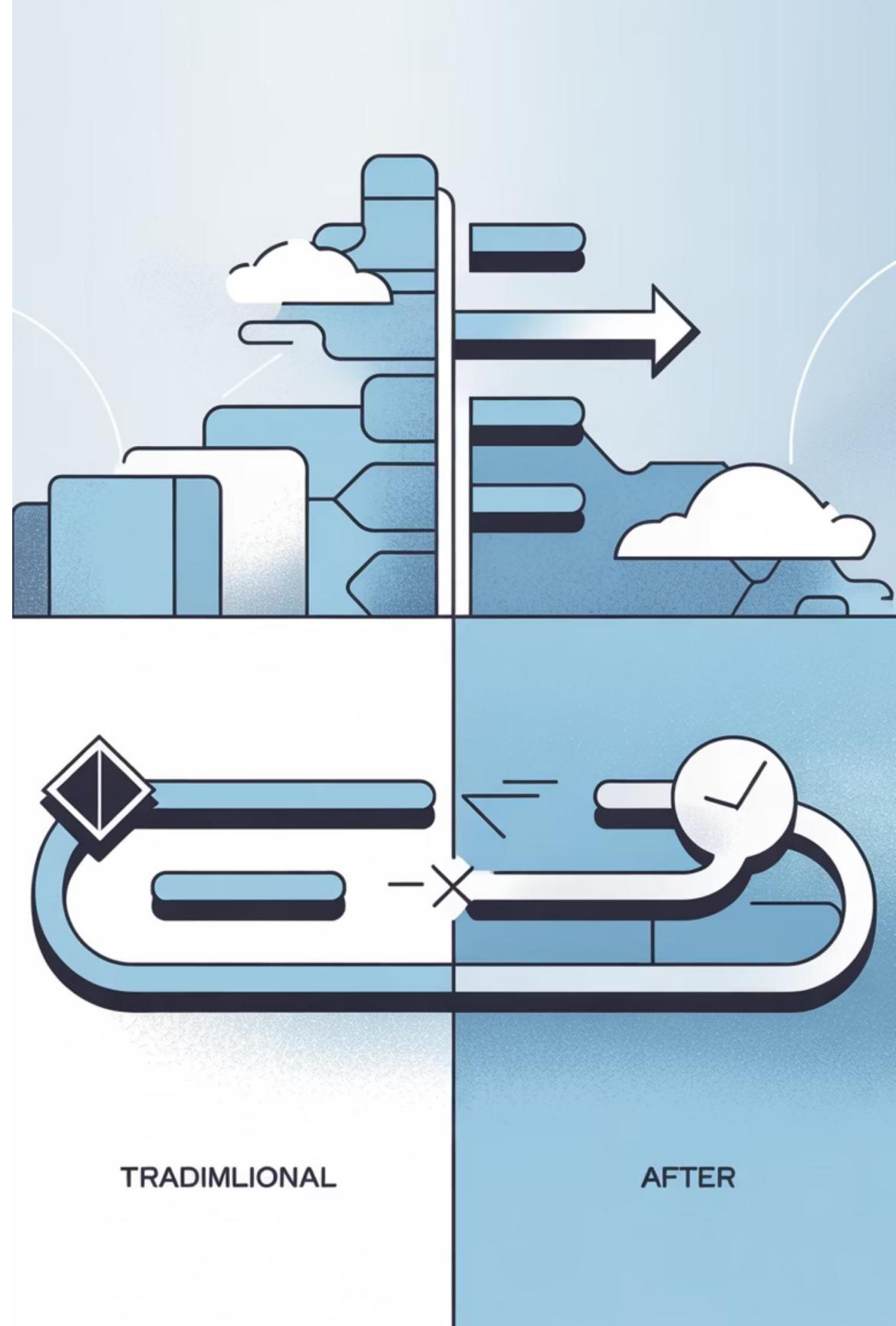
Revolutionary Development Approach

Old Way ✗

- Phone calls required
- Documentation requests
- Waiting for responses
- Manual integration

New Way ✓

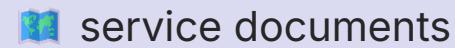
- Tools integration: metadata usage
- No phone calls: eliminated dependency
- No documentation: requests unnecessary
- Self-describing: service intelligence



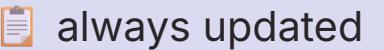
Self-Describing Services Advantage



Clear maps



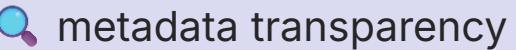
Built-in manual



Instant access

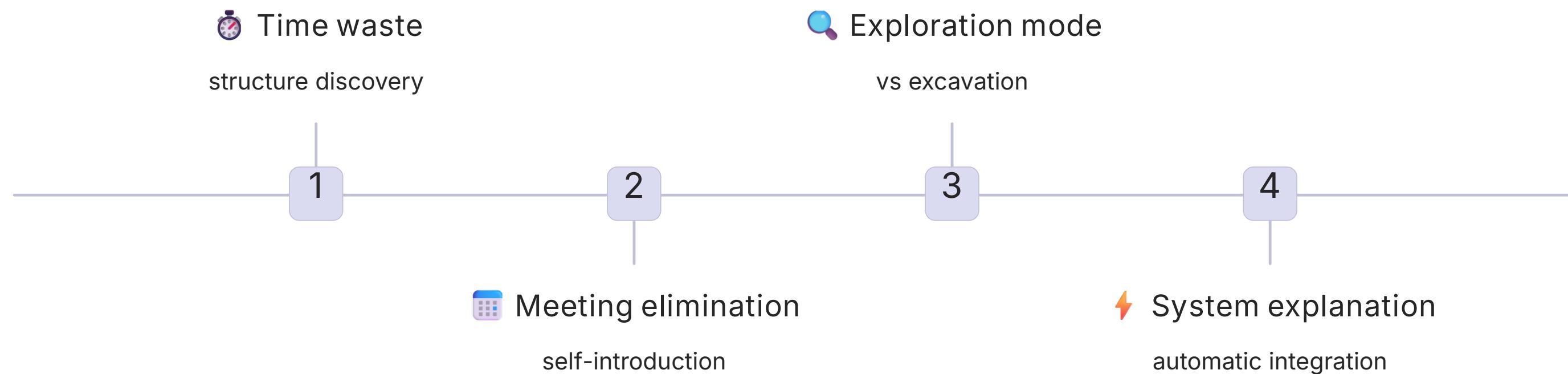


Complete structure





Development Process Transformation



Integration Paradigm Shift

🔍 Exploration approach

discovery-driven

🗣️ System communication

self-explanation

Because when systems can explain themselves

⛏️ Excavation elimination

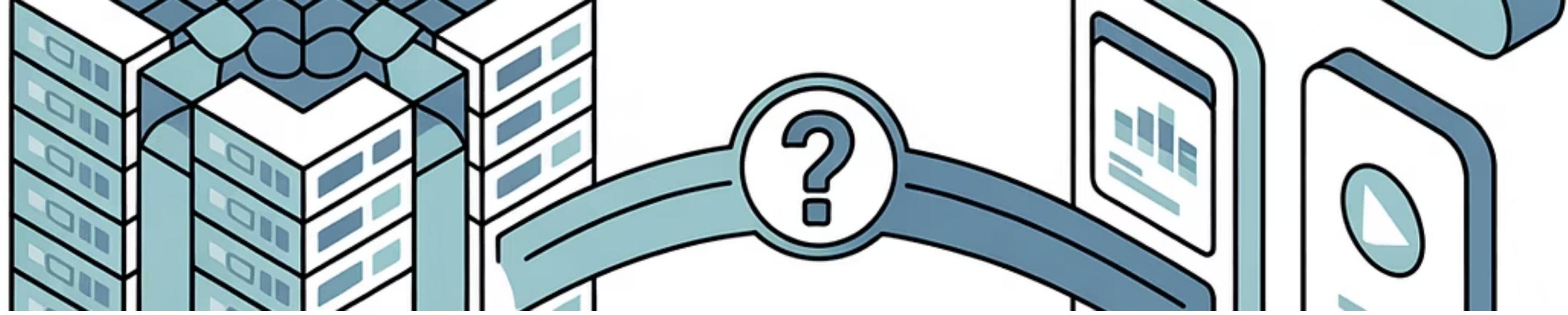
no digging

⚡ Effortless integration

natural process



SAP Gateway: Your Smart Bridge to SAP Data



The SAP Data Access Challenge

💪 SAP systems = powerful business data

📱 Modern apps = mobile, cloud, Fiori

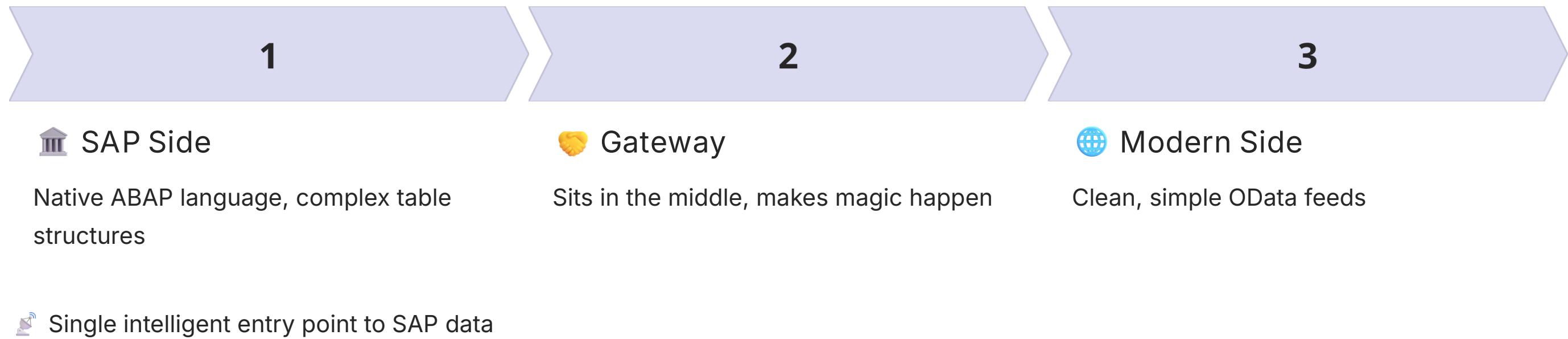
🚧 Getting data out = big challenge

😱 Complex integration requirements

How do you bridge powerful legacy systems with modern application needs?

SAP Gateway: Your Diplomatic Translator

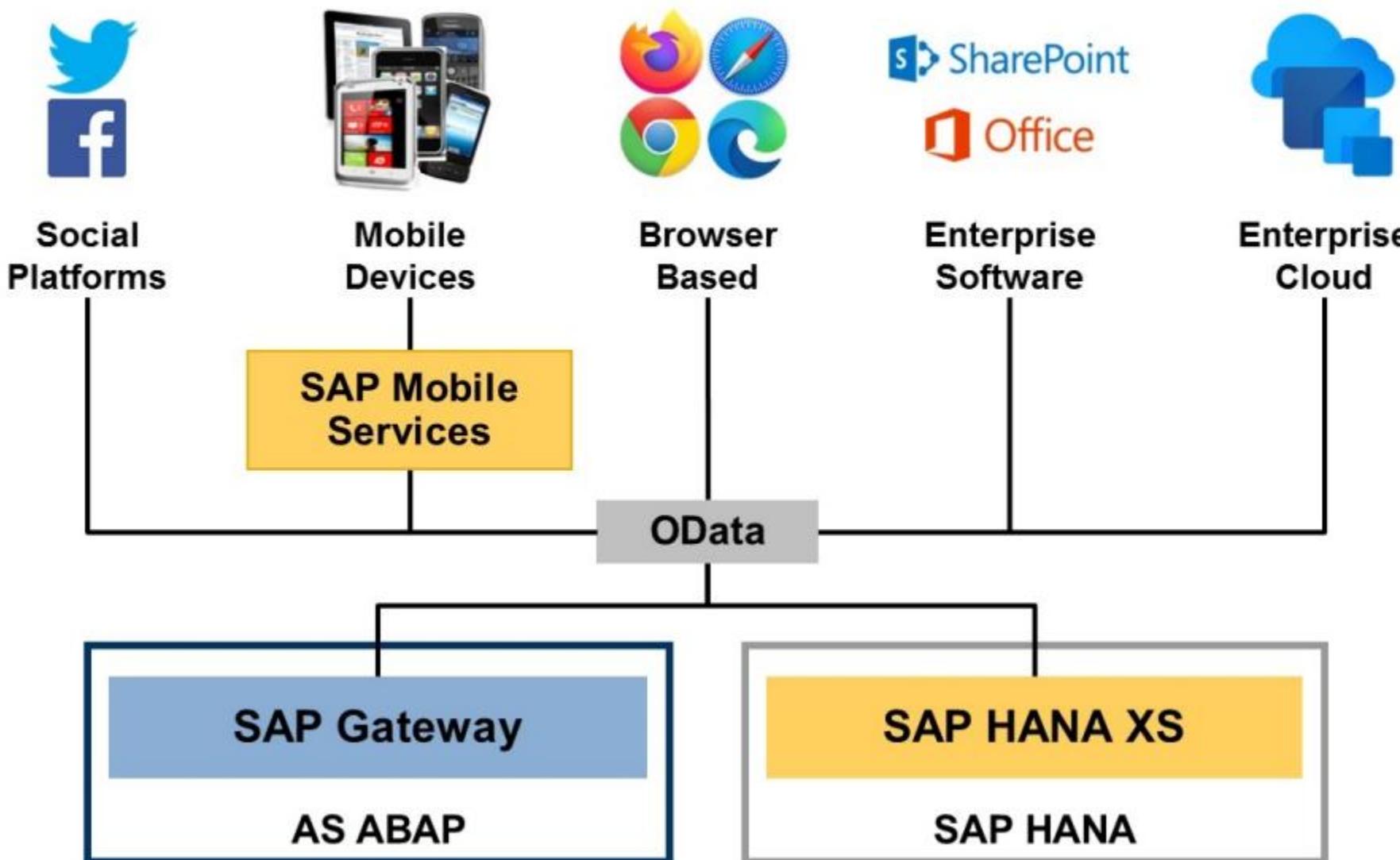
The Universal Translator Concept:



Translates complex SAP data into open
OData standard

SAP Gateway: Your Diplomatic Translator

The Universal Translator Concept:



Three Key Benefits That Change Everything

Gateway Transformation:

1



Simplified Access

No need to understand SAP internals

2



Broad Reach

Mobile, cloud, web, social platforms

3



Reduced Complexity

No specialized SAP knowledge required

External applications just speak OData - Gateway handles the rest

Lisa's Power BI Success Story

The Business Challenge:

-  Lisa needs real-time sales data for Power BI
-  SAP system seemed like a fortress
-  IT estimated 6 months for custom integration
-  Innovation blocked by integration complexity

Traditional approach: 6 months of
custom development

Gateway Solution: 3 Days Instead of 6 Months

The Gateway Approach:

01

-  Expose sales data as OData service

03

-  Total time: 3 days

02

-  Lisa connects Power BI directly to OData feed

04

-  Dashboard refreshes automatically every hour

Live SAP data, zero ABAP knowledge required

Mobile Enhancement: SAP Mobile Services

Enhanced Mobile Experience:



SAP Mobile Services adds to communication path



⚡ Optimized data delivery for mobile

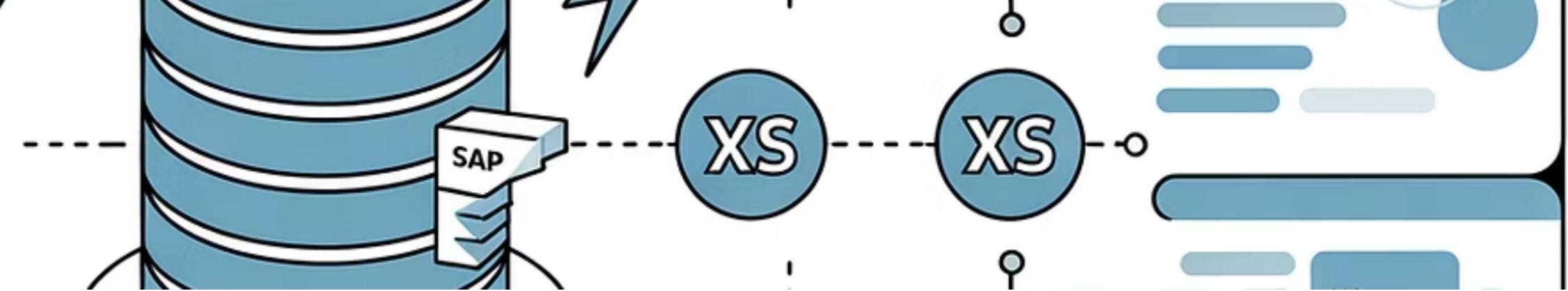


OFF Offline capabilities automatically



🔒 Security features built-in

Your mobile apps get enterprise-grade capabilities without custom development



SAP HANA XS: In-Memory Data Bridge

HANA-Specific Bridge:

- For data residing in SAP HANA
- XS provides same bridge functionality
- In-memory data accessible through web standards
- Same Gateway principles, HANA-optimized

Makes in-memory data
accessible through
modern web standards

Universal Adapter for the Digital World

The Digital Transformation Bridge:



🔌 Universal adapter concept



💪 Old, powerful SAP systems



🗣 Speaking language of modern digital world

🚫 No deep SAP expertise required for consuming apps

SAP Gateway removes friction, making
SAP data easily consumable

Innovation Unleashed

The Innovation Questions:

 How many projects delayed by SAP complexity?

 How many ideas never happen due to integration barriers?

 What if SAP data was as accessible as a web page?

 Innovation no longer limited by integration complexity

When your most valuable business data becomes **web-accessible**, everything changes

Your OData Services Control Center





Your Air Traffic Control Tower

Think of it like being the air traffic controller for all your data services



Robust management tools within SAP



Centralized visibility and control

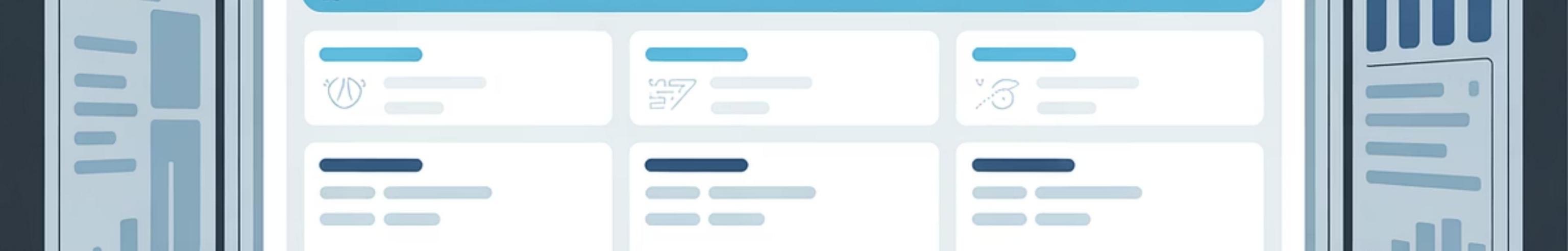


Dedicated transactions (apps within SAP)



Complete service lifecycle management

Even though OData makes consumption easy, management needs powerful tools



Transaction /WFND/MAINT_SERVICE

Your Mission Control

For OData version 2 services, you'll use transaction /WFND/MAINT_SERVICE

Central Hub for OData v2

- Manages all registered OData services
- Three organized operational areas
- Add, delete, configure services
- Complete visibility from one location

Your central hub for managing all registered OData services

Three Clear Operational Areas

It's divided into three clear areas that make everything organized

01

Service Catalog

List all services, descriptions, settings

02

ICF Nodes

Internet Communication Framework maintenance

03

System Aliases

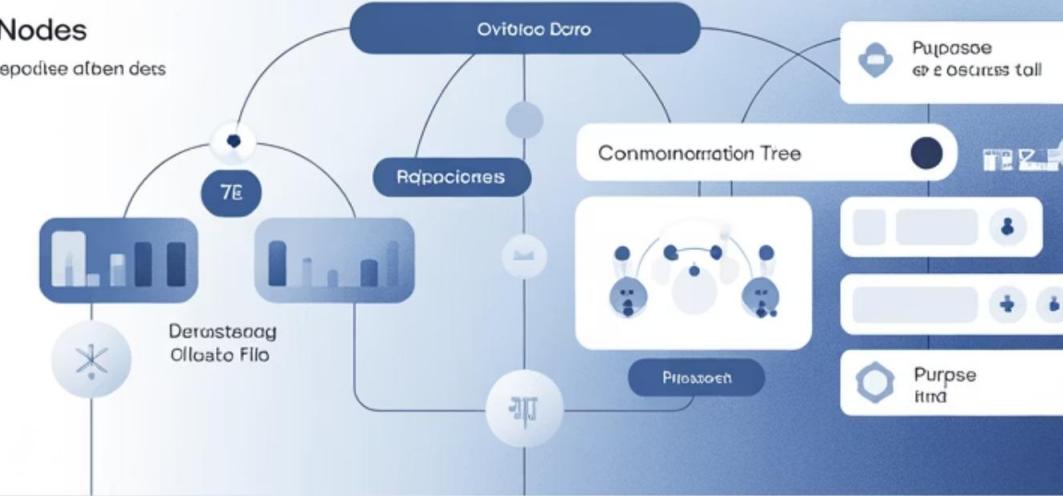
Multi-system connection management

Each area handles specific aspects of service lifecycle management



ICF Nodes

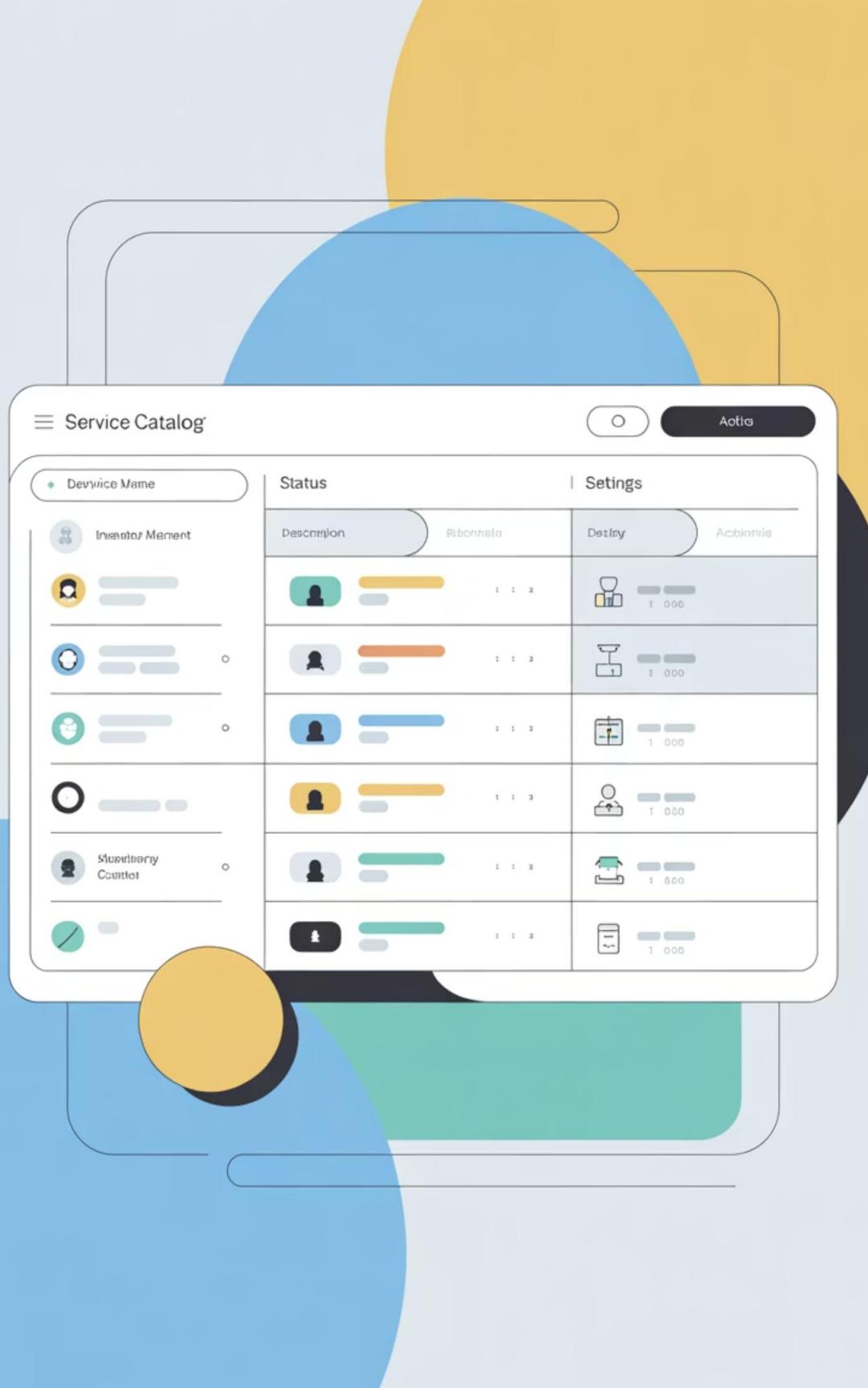
Connopodote alben dete eotla



System Aliases

Connopodote alben dete eotla





Service Catalog: Your Service Inventory

First, the Service Catalog. Here you'll find a list of all your services

Complete Service Overview

- List of all services with descriptions
- Various settings and configurations
- Add new services
- Delete existing services
- View detailed service information

Your comprehensive inventory system for all OData services

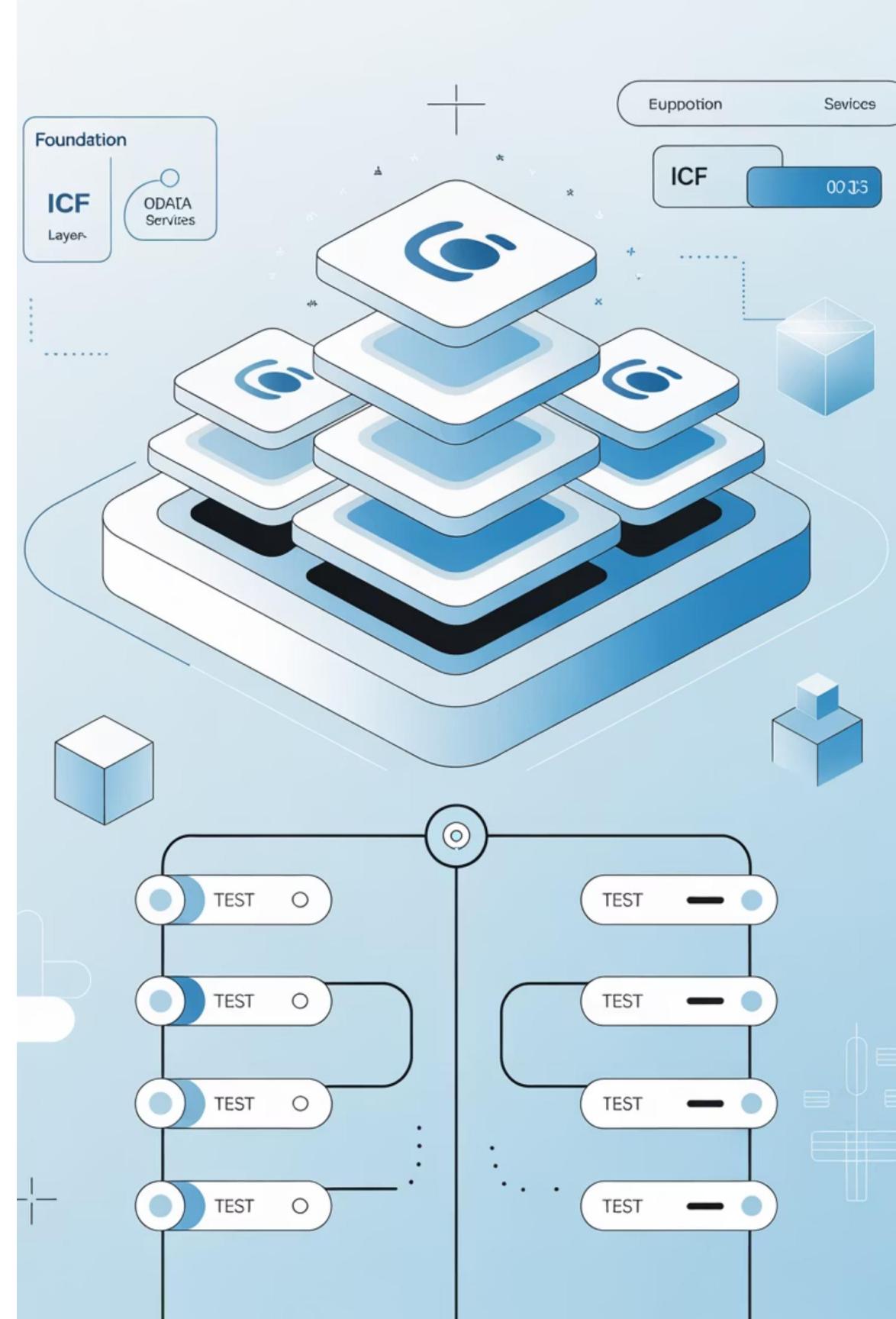
ICF Nodes: Communication Foundation

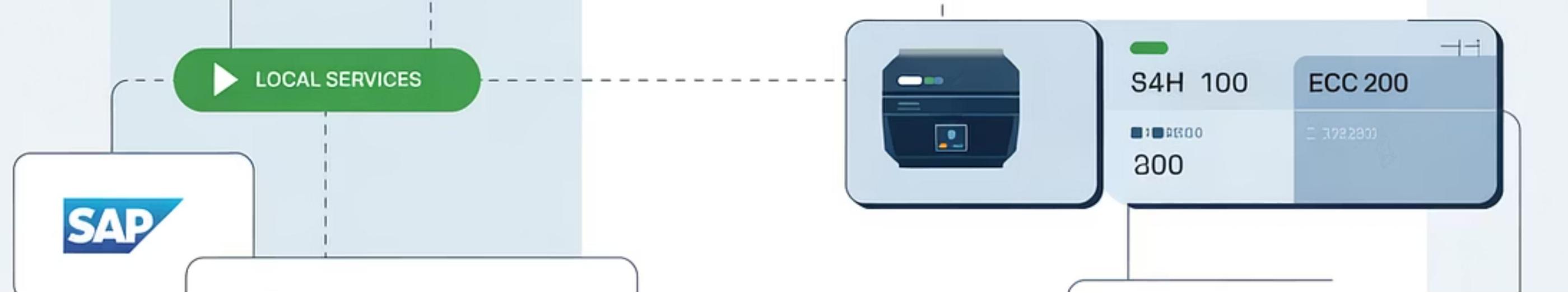
Second, ICF Nodes. This section allows you to maintain and test

Web Communication Management

- Internet Communication Framework services
- Maintain and test communication services
- Fundamental for web-based communication
- Verify connectivity and functionality

The foundation layer that makes web-based communication possible





System Aliases: Multi-System Connections

Third, system aliases. This is where you manage connections to other SAP systems

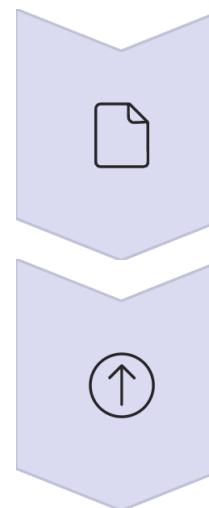
Cross-System Data Management

- Manage connections to other SAP systems
- Services pulling data from multiple sources
- "Deployed only" = no external system needed
- No system alias assigned for local services

Essential when OData services need data from multiple SAP systems

OData Version 4: Split Administration

For OData version 4 services, things work a bit differently



/IWBEV4_ADMIN

Define service groups

/IWFNDV4_ADMIN

Publish service groups

- Split administration across two transactions
- Fiori Launchpad notification service was first productive v4 service

OData version 4 administration splits into definition and publication phases

Robert's Operations Transformation

Take Robert from the IT operations team. He manages over 30 OData services

Before Centralized Tools

- 30+ OData services across SAP modules
- Checking multiple systems and log files
- Hours to track service status
- Scattered information, inefficient troubleshooting

Tracking service status meant checking multiple systems and log files



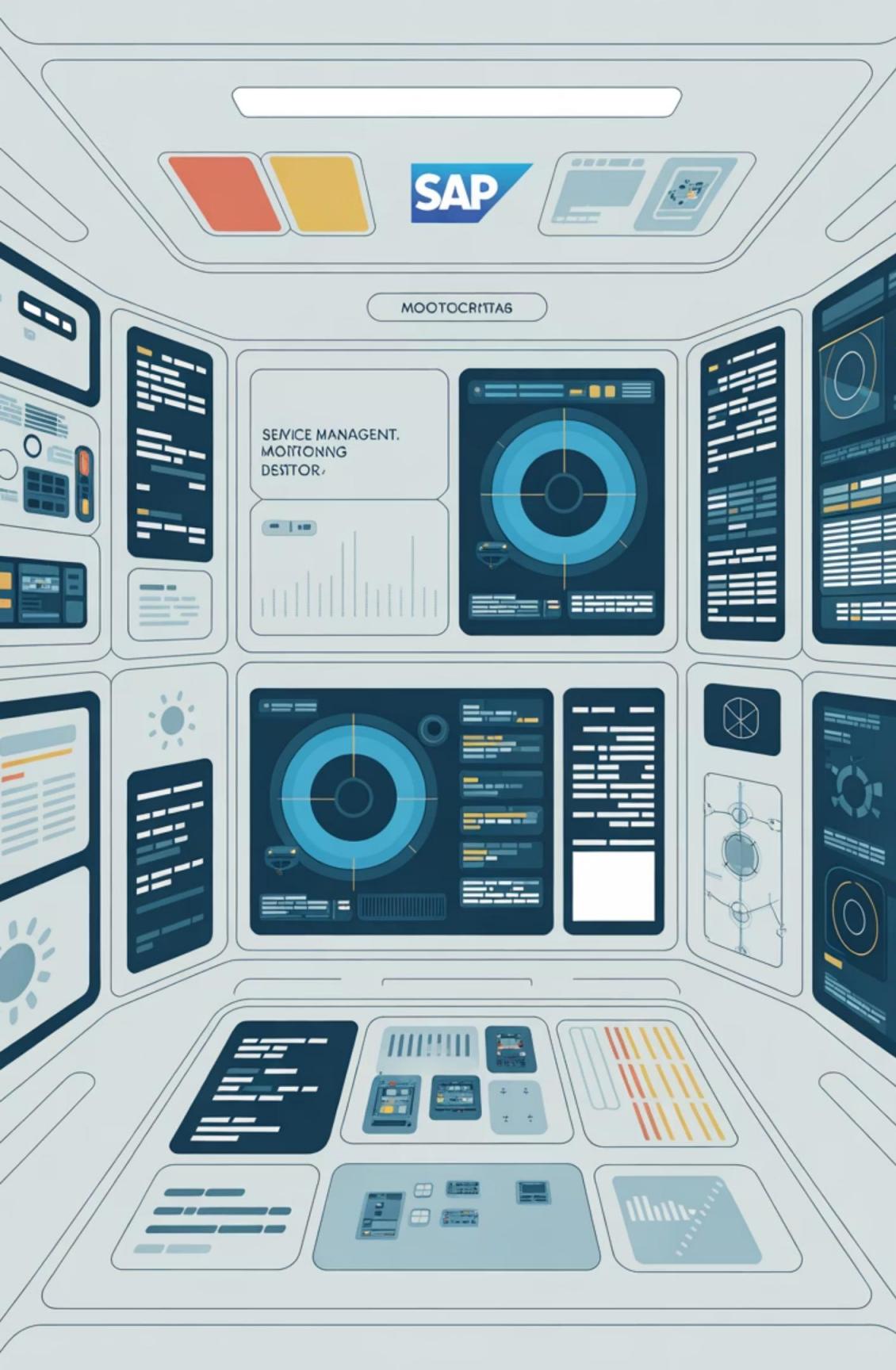
Robert's New Reality: One Transaction, Everything Visible

Now he opens one transaction and sees everything

-  Which services are active
-  Which ones need updates
-  Where performance bottlenecks are
-  New customer data feed: minutes instead of hours

When sales team requests new data feed, he configures it in minutes





Integrated Mission Control

The power is in the integration. These aren't separate tools you need to learn

- Same login, interface patterns, security model
- Test services within SAP environment
- Monitor performance without leaving SAP
- Troubleshoot issues in familiar interface

Like having mission control perfectly integrated with the spacecraft



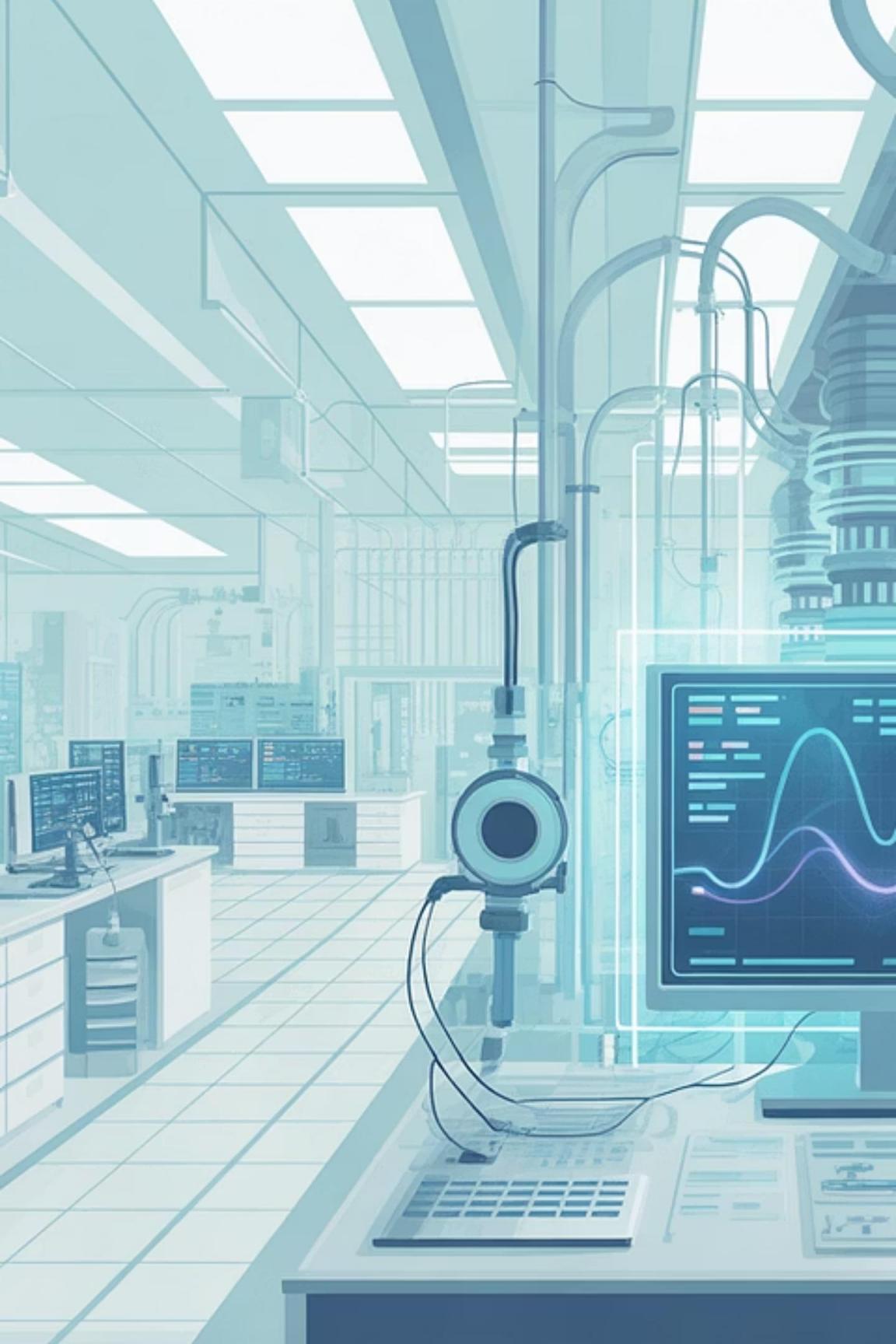
Focus Shift: Administration to Innovation

Think about your current service management process

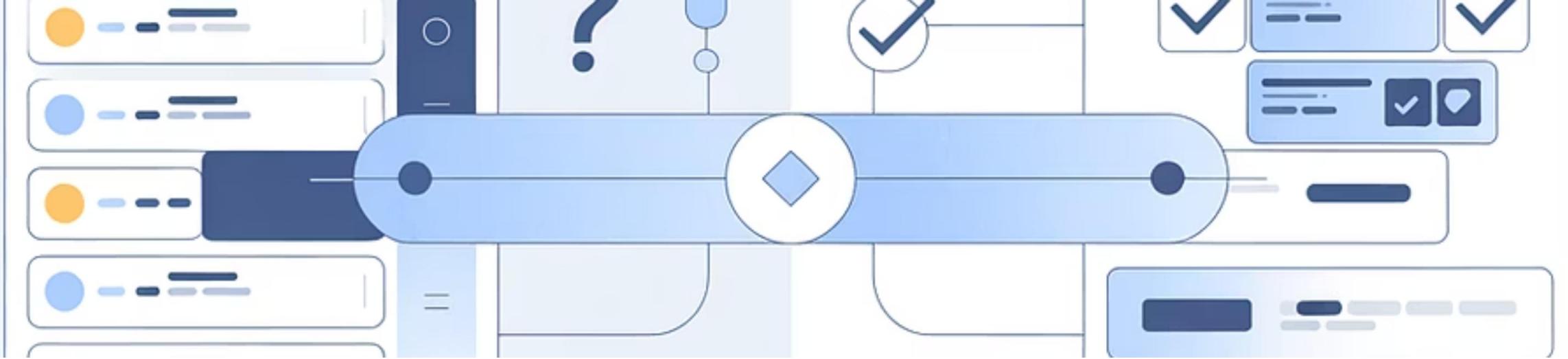
The Transformation Questions

- How much time spent tracking service issues across systems?
- How much easier with everything in one location?
- Team focus: innovation instead of administration
- Service management as intuitive as the services themselves

When management becomes intuitive, teams focus on innovation instead of administration



Testing Your OData Services:



How Do You Know It's Working?

The Critical Question

Configuration without testing is just hope, not certainty

- ⚙️ OData service configured
- ❓ Is it working correctly?
- 🖥️ You need to test it
- 🔐 SAP provides the perfect tool

SAP Gateway Client: Your Personal Laboratory

Transaction /IWFND/GW_CLIENT



Personal laboratory for data services



Test all OData service functionalities



Directly from within SAP



No external tools required

Your dedicated testing environment for comprehensive OData validation



Simple Data Retrieval Testing

Basic Read Operations

01

 Simple data retrieval = read request

02

 Enter OData service URL

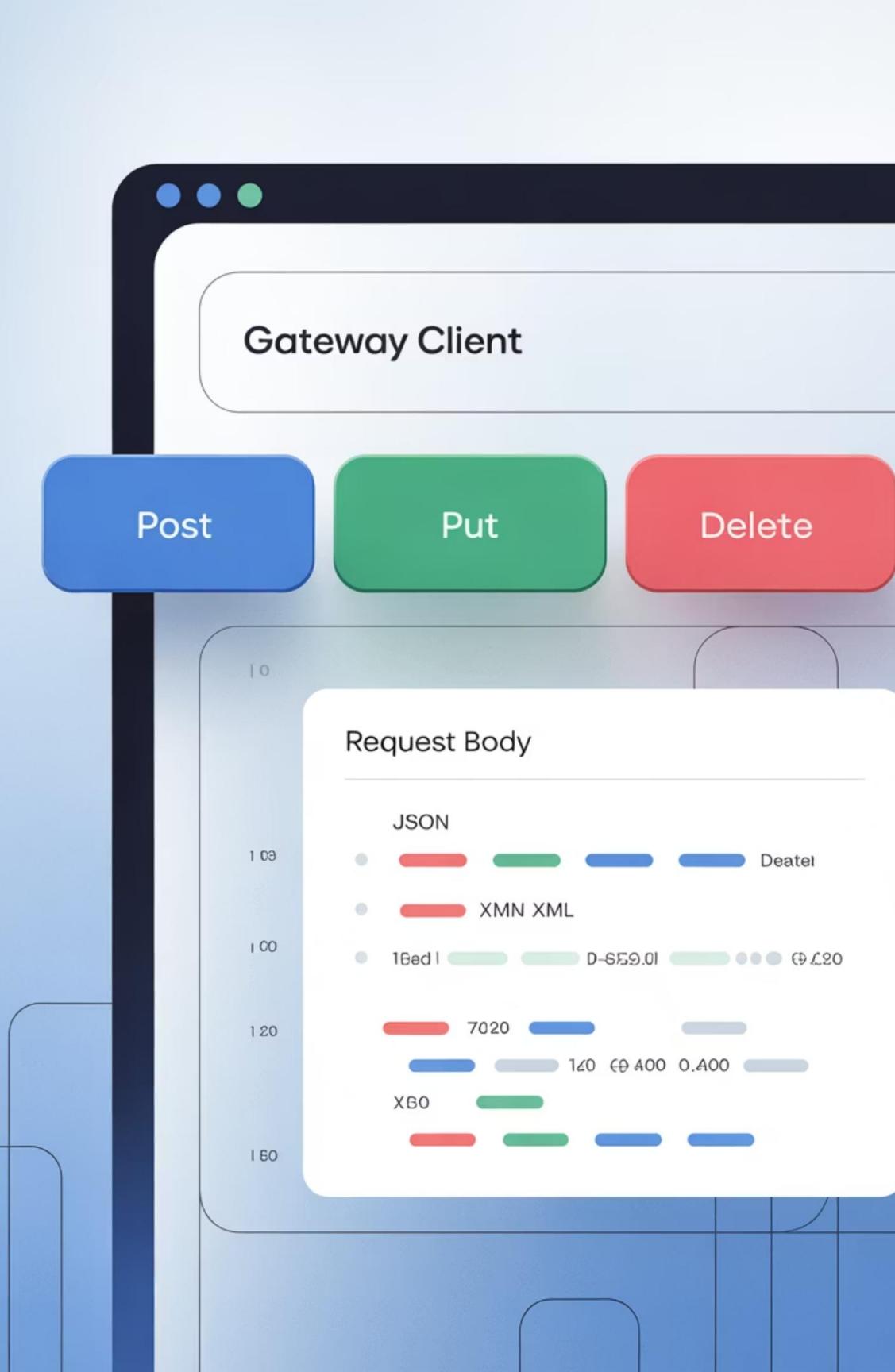
03

 Execute the request

04

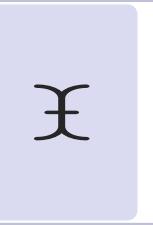
 Immediate results display

Just enter the OData service URL and execute - that simple



Complex Operations: Create, Update, Delete

Advanced Testing Capabilities

-  Creating data
POST requests with request body
-  Updating data
PUT requests with modifications
-  Deleting data
DELETE requests
-  Request body
Prepare request body with data to send

Complex operations require additional adjustments but follow same testing pattern

Save and Reuse: Regression Testing Made Easy

Testing Efficiency Features

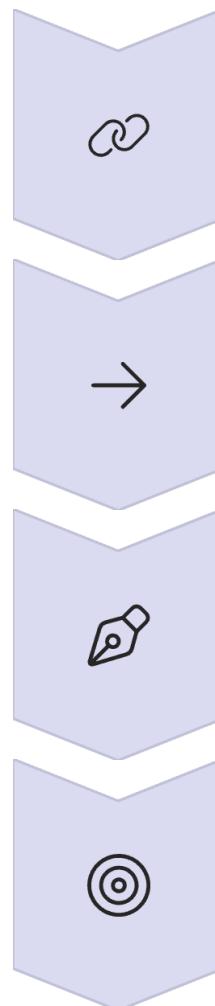
[AUDIO CUE: "The best part? You can save all your adjustments and test cases"]

-  Save all adjustments and test cases
-  Reuse for future testing
-  Regression testing made simple
-  Build library of test scenarios

Making regression testing a breeze with saved test cases

Integrated Workflow: No Context Switching

Seamless Integration Benefits



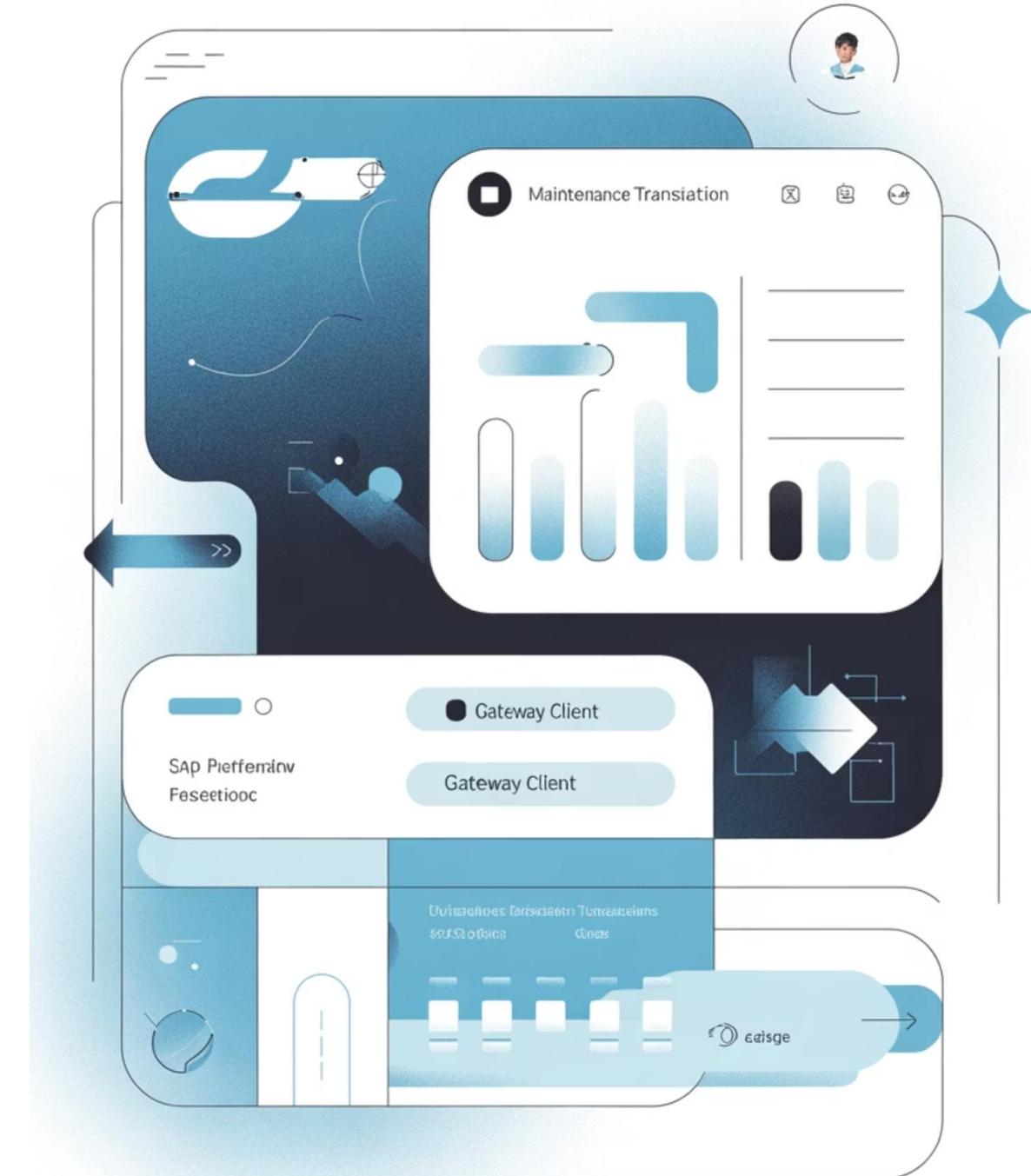
All Gateway transactions interconnected

Jump between service maintenance and client

Streamlined workflow

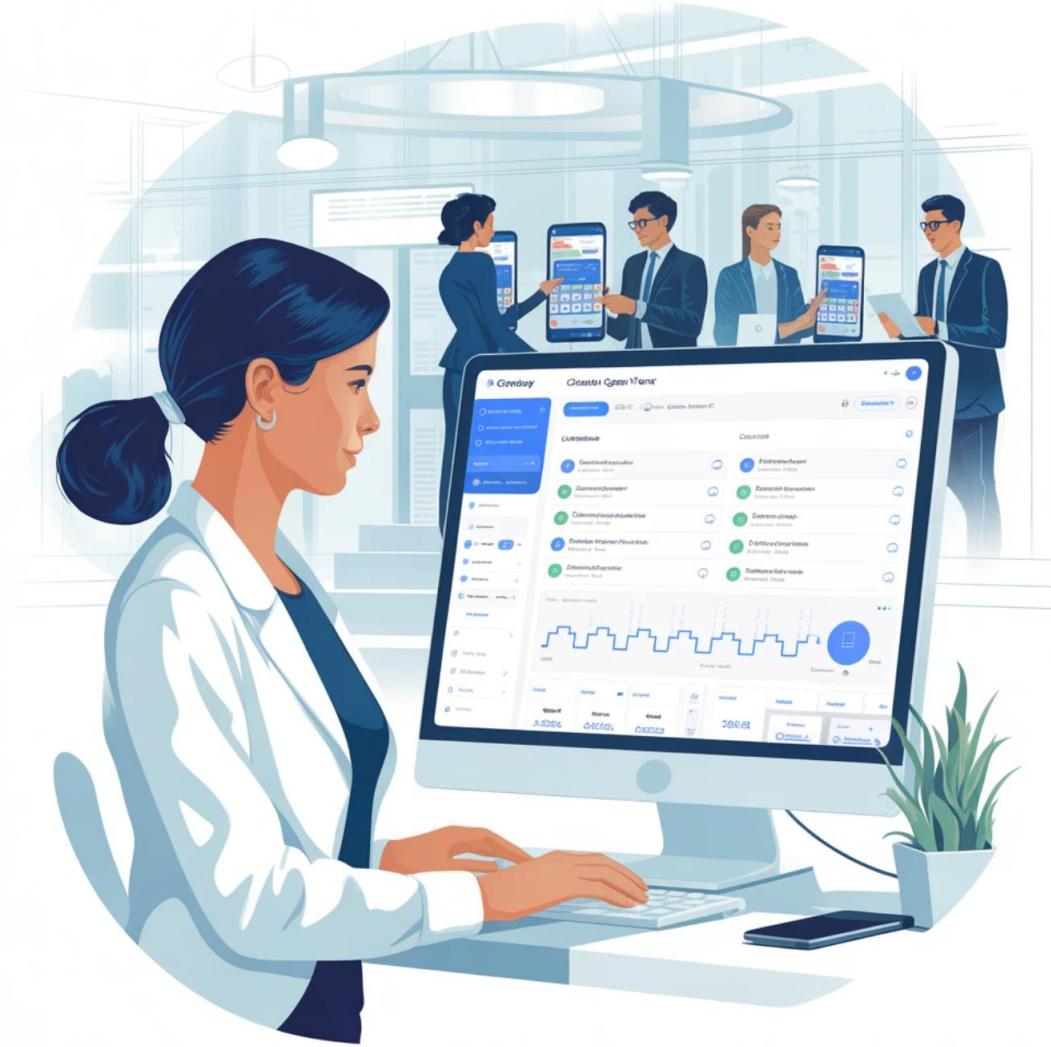
No switching applications or learning new interfaces

You're not switching between different applications - it's all integrated



Amanda's QA Success Story

The QA Challenge

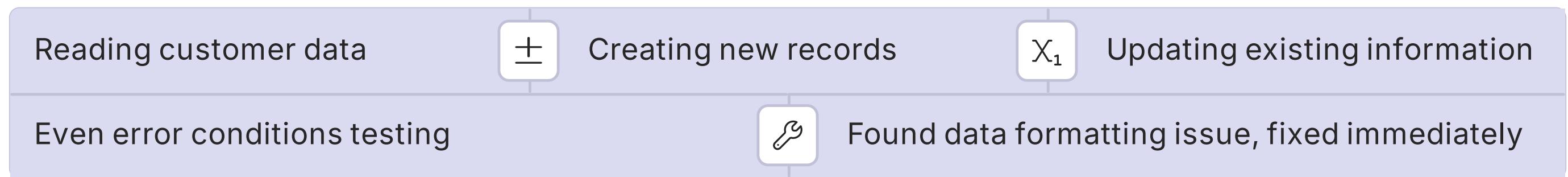


- Amanda from Quality Assurance team
- Validate new customer service before mobile development
- No waiting for developers to build test interfaces
- Used Gateway Client for comprehensive testing

Instead of waiting for developers, she used the Gateway Client

Comprehensive Testing: Every Scenario Covered

Amanda's Testing Approach



When she found issues, she jumped to maintenance, fixed, and retested immediately

The Game-Changing Workflow

Integrated Testing Cycle

01

Configure service in maintenance transaction

02

Jump to Gateway Client to test

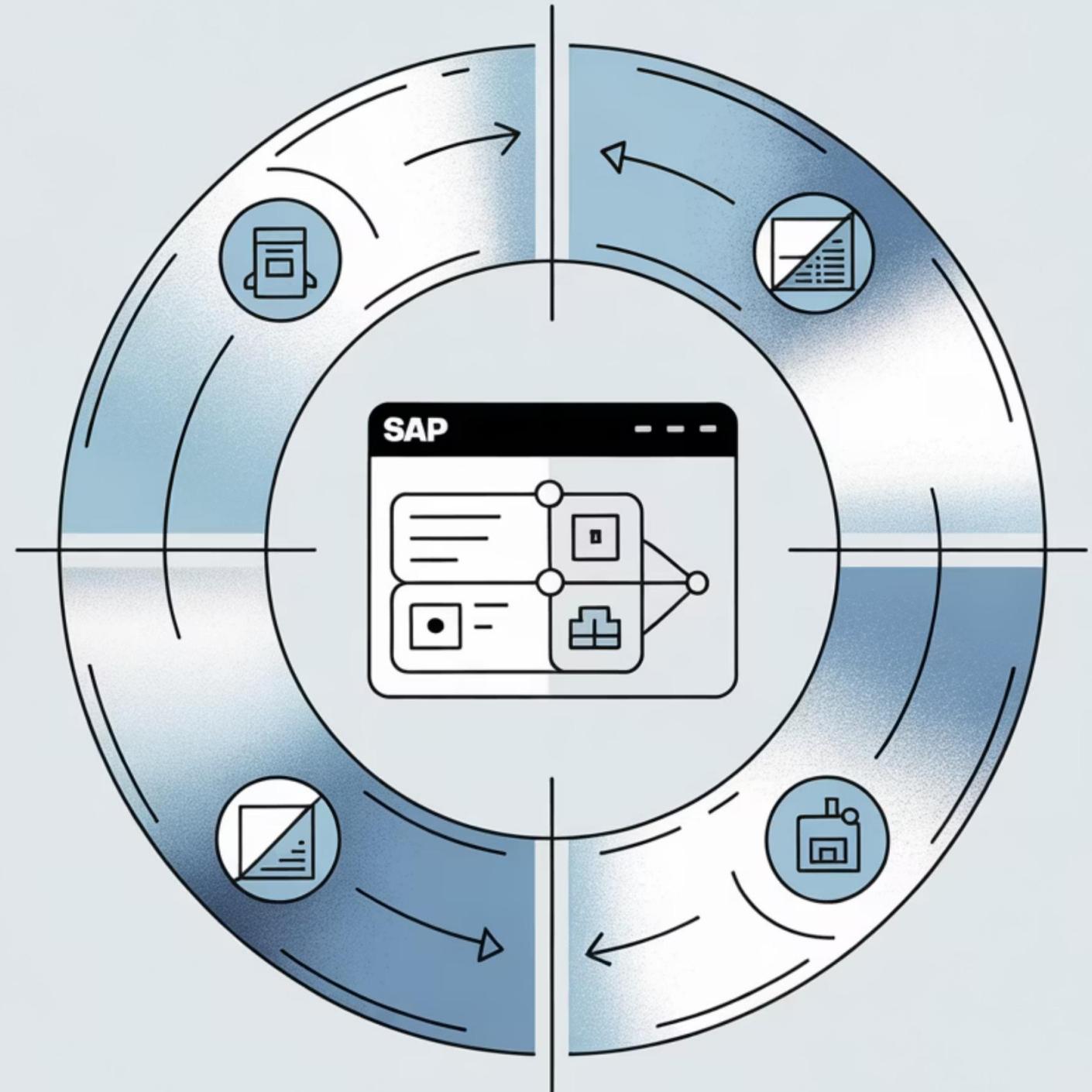
03

Find issue? Jump back to maintenance

04

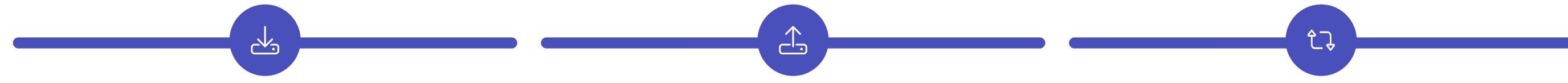
Fix it, then retest immediately

All within same SAP session - no external tools,
no waiting



Comprehensive Testing Capabilities

Full HTTP Method Coverage



GET requests

Retrieve data

POST requests

Create new records

PUT requests

Update existing data

DELETE requests

Remove records

Save & Rerun

Each test saved and rerunnable

Every HTTP method supported with save and rerun capabilities

Flight Simulator for Data Services

Safe Testing Environment

-  Flight simulator concept for data services
-  Test every scenario safely
-  Before going live with applications
-  Risk-free experimentation environment

You can test every scenario safely before going live

Quality: Inevitable, Not Accidental

The Testing Transformation

Before

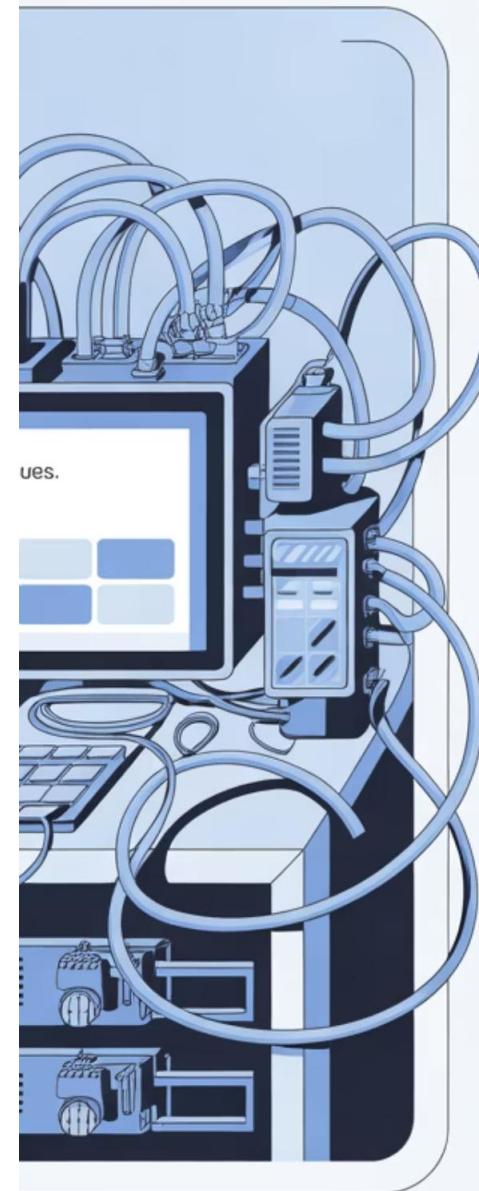
- ⌚ How much time building custom test interfaces?
- 🐛 How many issues caught earlier with simple testing?

After

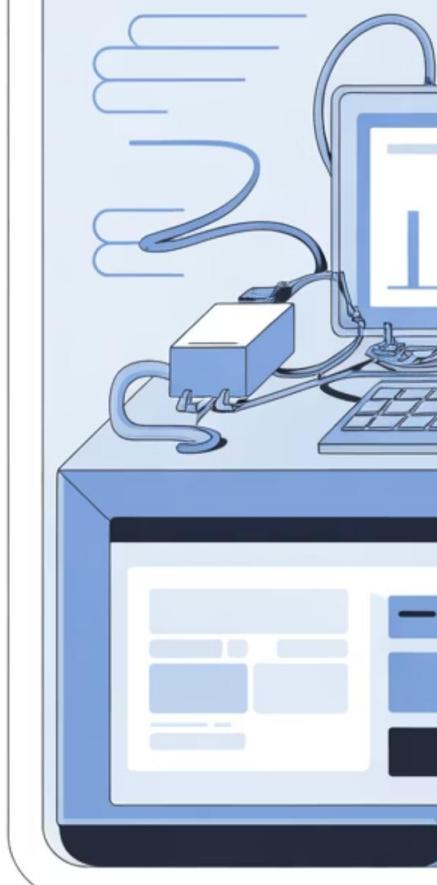
- ⌚ Testing becomes effortless
- ✨ Quality becomes inevitable instead of accidental

When testing becomes effortless, quality becomes inevitable

Comstom Test Test
oscldoon



Gateway
Client-quality
outcomes.



Crafting Your Data Wishlist in a URL

Your Very Specific Data Wishlist

One of the coolest things about OData

Your Very Specific Data Wishlist

The Power of Precision:

-  Ask for exactly what you need
-  Simply construct a web address (URL)
-  Like writing a specific data wishlist
-  No more "surprise me" data requests

It's like writing a very specific data wishlist in a URL

Think of it like ordering at a restaurant

Restaurant Ordering vs. Data Requests

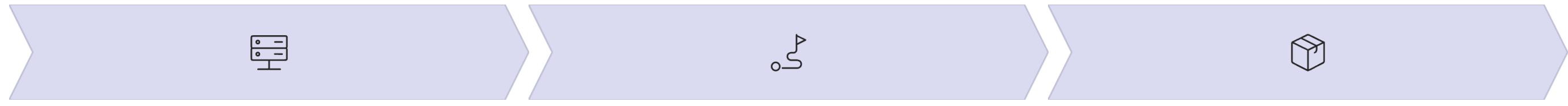
Precise Ordering Analogy:

-  Restaurant: "Surprise me" vs. specific order
-  OData: Specify exactly what you want
-  How much data you need
-  How it should be prepared/formatted

You specify exactly what you want, how much, and how it should be prepared

An OData request is built from several parts

OData URL Building Blocks



Server Address

Base URL (<https://server:port>)

Service URL

Path to specific OData service

Entity Set

Collection of data you want to access

- Example: /sap/opu/odata/iwbep/gwsample_basic/ProductSet

Here's where the real power comes in

Query Options: The Real Power



Filter

Which records to include



Select

Which fields to return



Format

JSON, XML, etc.



Top

Limit number of results

Add parameters to filter, select fields, order results, and much more

Question mark dollar sign format equals JSON

Query Options in Action

Practical Query Examples:

-  ?\$format=json - Data in JSON format Reducing the amount of data transferred with precise field selection
-  &\$top=3 - Only first three results
-  &\$select=ProductID,Category,Name - Specific fields only

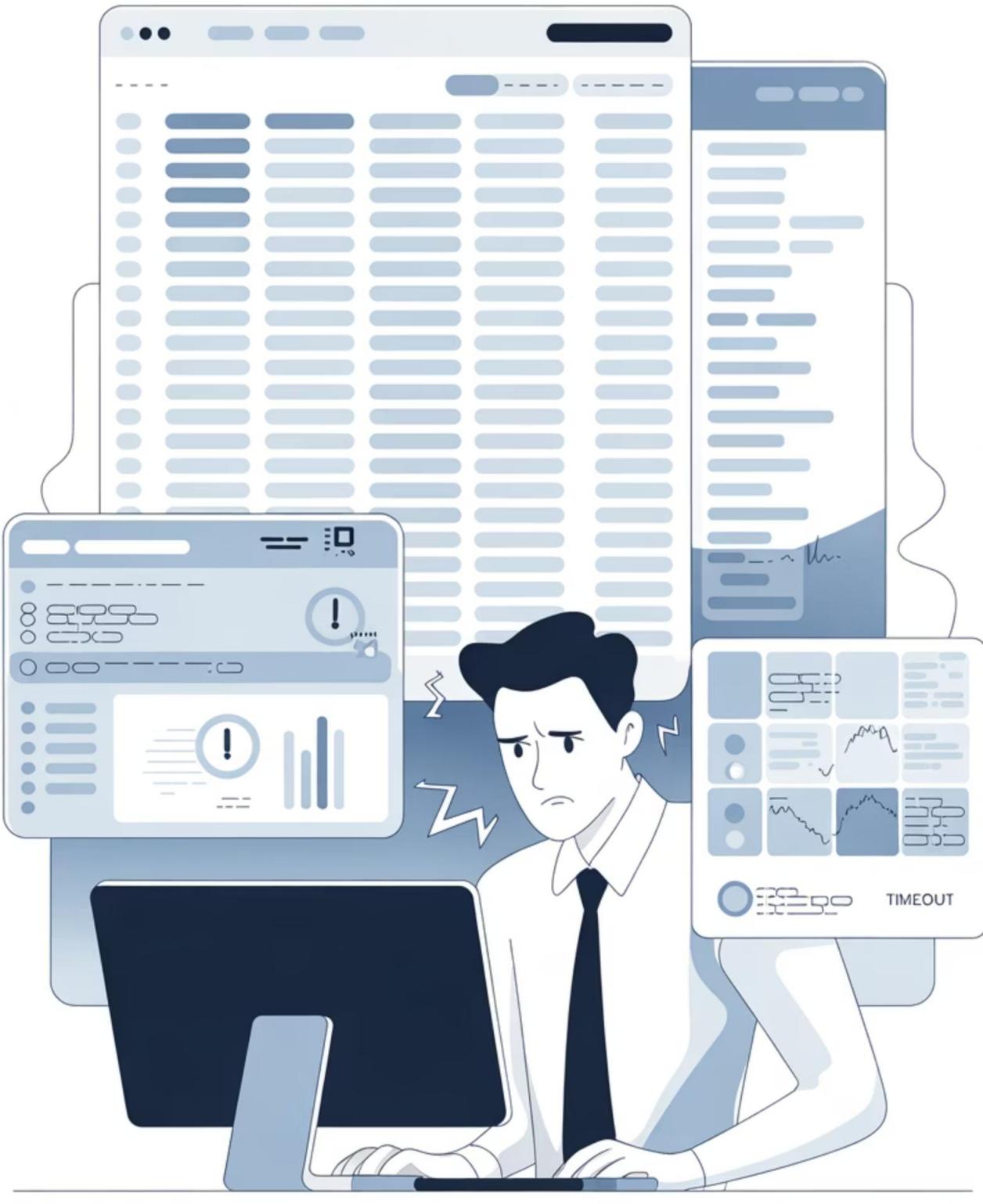
Take Kevin from Business Intelligence

Kevin's BI Performance Breakthrough

The Performance Problem:

-  Kevin from Business Intelligence team
-  Needed product data for dashboard
-  Full product records: 200+ fields each
-  Initial query timing out and crashing reports

Massive records with over 200 fields were timing out his reports



Using OData query options, he crafted a precise request

Kevin's Precision Solution



\$select - Only 6 fields needed



\$filter - Only active products



\$top - Limit to 50 records per page



Result: Timeout to 2 seconds completion

His query went from timing out to completing in under two seconds

There are many more query options available

Revolutionary Query Options Arsenal



\$orderby

Sort results



\$skip

Paginate through data



\$expand

Include related data



Full documentation

Available at odata.org

Incredibly flexible and precise data requests with extensive options

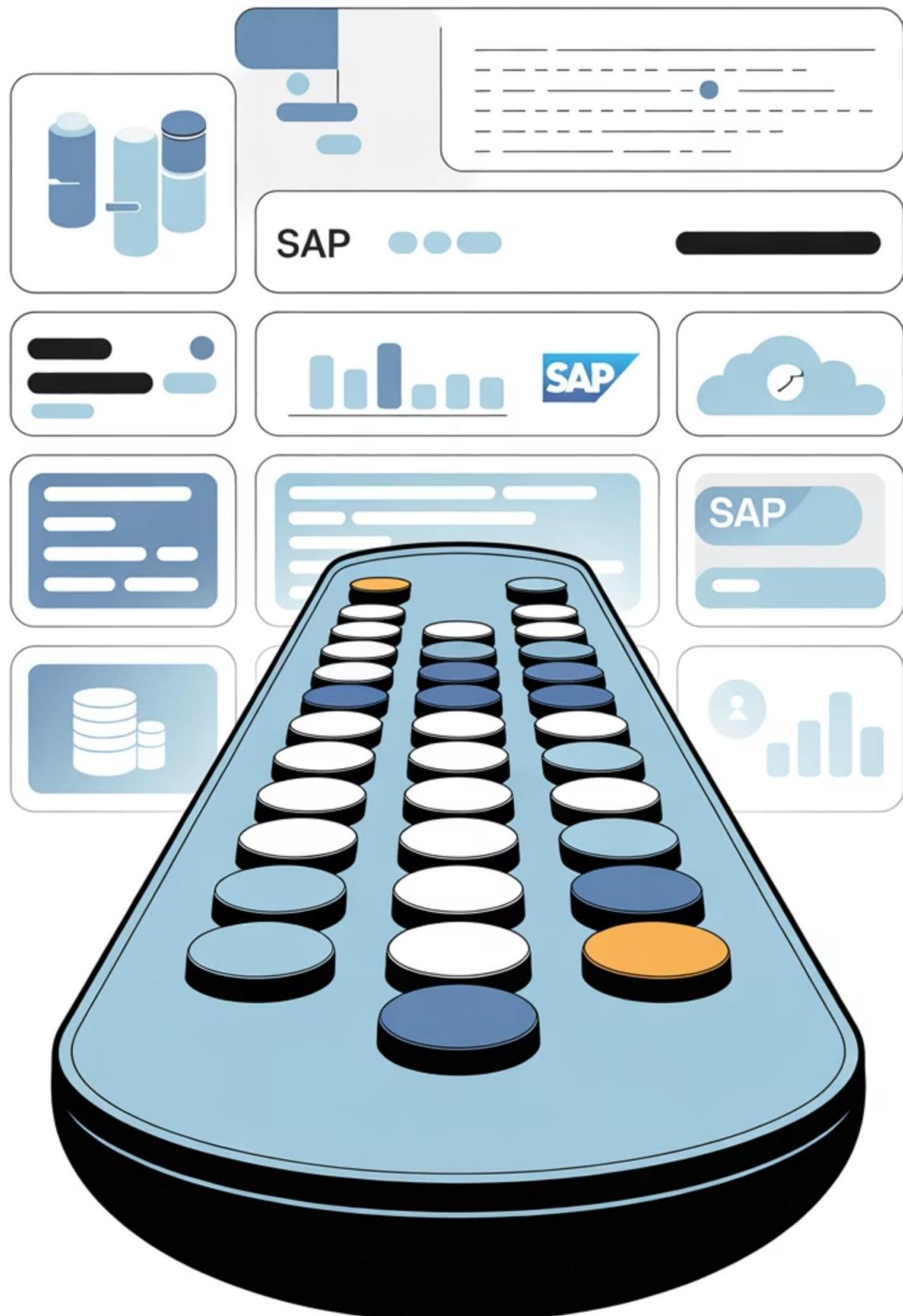
The beauty is in the readability

Human-Readable Data Conversations

Conversational Data Requests:

- Anyone can understand OData URLs
- Reads like a conversation
- "Give me products, ID and name only, first 10, as JSON"
- No complex query languages to learn

No database-specific syntax to memorize - it reads like a conversation



It's like having a universal remote control

Universal Remote Control for Data



Universal remote control concept



Point, click, get exactly what you need



Works with any OData source



Powerful, human-readable URLs

Point, click, and get exactly what you need from any data source

Think about your current data requests

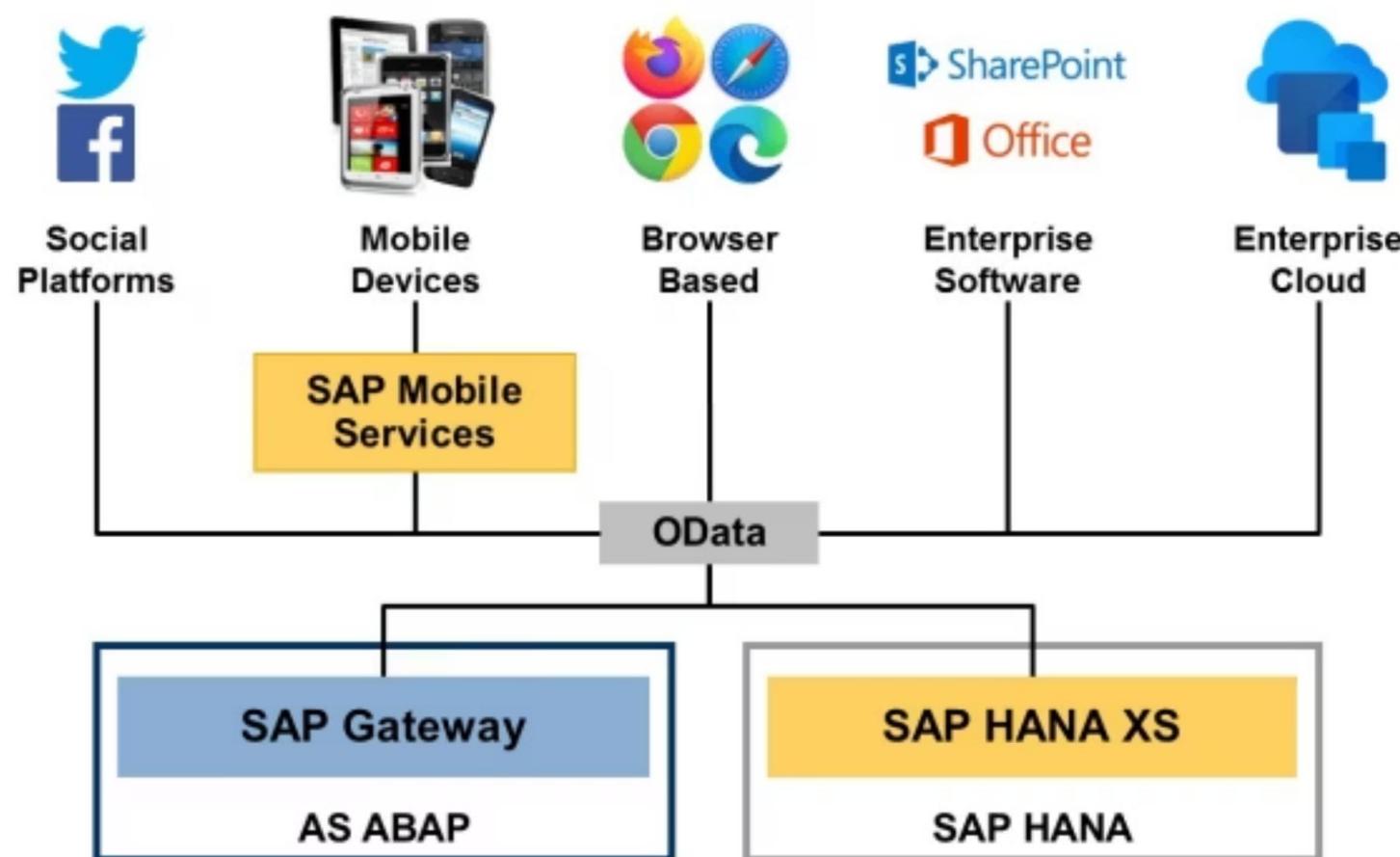
Surgical vs. Wholesale Data Requests

The Performance Revolution:

- 🤯 How much unnecessary data are you downloading?
- 🚶 How much network bandwidth could you save?
- 🌐 Surgical requests vs. wholesale data dumps
- ⚡ Performance becomes advantage, not limitation

When every data request becomes surgical,
performance stops being a limitation

SAP Gateway Service Management Control Center



Production Crisis: Five Services, Three Problems

The Nightmare Scenario:

- ⚠ One service returns errors
- 🐌 Another runs painfully slow
- 🔌 Third cannot connect to backend
- 😢 Users frustrated, no visibility into root cause

You need a control center to manage this chaos

The screenshot shows the SAP Activate and Maintain Services interface. The top navigation bar displays the title "/IWFND/MAINT_SERVICE". The main area is titled "Service Catalog" and contains a table listing various services. The columns include Type, Technical Service Name, Service Description, External Service Name, Namespace, OAuth..., Soft State Status, and Processing Mode. Below the catalog, there are two tabs: "ICF Nodes" and "System Aliases". The "ICF Nodes" tab shows a table with columns: Status, ICF Node, Session Time-out, Soft State, and Description. The "System Aliases" tab shows a table with columns: SAP System Alias, Description, Default System, and Metadata.

Type	Technical Service Name	V...	Service Description	External Service Name	Namespace	OAuth...	Soft State Status	Processing Mode
BEP	GFD_CONFIG_SRV	1	Generic Form Designer Configuration details	GFD_CONFIG_SRV		<input checked="" type="checkbox"/>	Not Supported	Co-deployed only
BEP	GHO_ODATA_SRV	1	OData Project for Global Hydrocarbon Operations	GHO_ODATA_SRV		<input type="checkbox"/>		Co-deployed only
BEP	IWFND/GWDEMO_SP2	1	ZCL_ZTEST_GWDEMO_DPC_EXT	GWDEMO_SP2	/IWBEPI/	<input type="checkbox"/>		Co-deployed only
BEP	ZGWSAMPLE_BASIC	1	SAP Gateway Sample Service - Basic	GWSAMPLE_BASIC	/IWBEPI/	<input type="checkbox"/>		Co-deployed only
BEP	ZHCMFAB_BEN_ENROLLMENT_SRV	1	HCM Benefits Enrollment	HCMFAB_BEN_ENROLLMENT_SRV		<input type="checkbox"/>		Co-deployed only
BEP	ZHCMFAB_COMMON_SRV	1	OData Service for reusable parts	HCMFAB_COMMON_SRV		<input type="checkbox"/>		Co-deployed only
BEP	ZHCMFAB_EMPLOYEELOOKUP_SRV	1	Employee Lookup	HCMFAB_EMPLOYEELOOKUP_SRV		<input type="checkbox"/>		Co-deployed only
BEP	ZHCMFAB_LEAVE_REQUEST_CR_SRV	1	HCM Leave Request Create - Fiori	HCMFAB_LEAVE_REQUEST_CR_SRV		<input type="checkbox"/>		Co-deployed only
BEP	ZHCMFAB_MY_EXT_ORG_SRV	1	My External Organizations	HCMFAB_MY_EXT_ORG_SRV		<input type="checkbox"/>		Co-deployed only
BEP	ZHCMFAB_MY_MEDICAL_INFO_SRV	1	My Medical Information	HCMFAB_MY_MEDICAL_INFO_SRV		<input type="checkbox"/>		Co-deployed only
BEP	ZHCMFAB_MYADDRESSES_SRV	1	HR Fiori Service: IT0006 Addresses	HCMFAB_MYADDRESSES_SRV		<input type="checkbox"/>		Co-deployed only
BEP	ZHCMFAB_MYBANKDETAILS_SRV	1	Bank details	HCMFAB_MYBANKDETAILS_SRV		<input type="checkbox"/>		Co-deployed only

Status	ICF Node	Session Time-out	Soft State	Description
OK	ODATA	00:00:00		Standard Mode

SAP System Alias	Description	Default System	Metadata

Your OData Restaurant Kitchen

Multiple Chefs, Multiple Dishes:

- Each chef = OData service
- Specific recipes = service configurations
- Different dishes = data responses
- Head chef needs central command station

SAP provides exactly this kind of control center

The screenshot shows the SAP Activate and Maintain Services interface with the title bar /IWEND/MAINT_SERVICE. The main area is divided into three sections: Service Catalog, ICF Nodes, and System Aliases.

Service Catalog: A grid of service entries. The columns include Type, Technical Service Name, Service Description, External Service Name, Namespace, OAuth, Soft State Status, and Processing Mode. Some entries have checkboxes for OAuth and Soft State Status.

Type	Technical Service Name	Service Description	External Service Name	Namespace	OAuth	Soft State Status	Processing Mode
BEP	GFD_CONFIG_SRV	1 Generic Form Designer Configuration details	GFD_CONFIG_SRV		<input checked="" type="checkbox"/>	Not Supported	Co-deployed only
BEP	GHO_ODATA_SRV	1 OData Project for Global Hydrocarbon Operations	GHO_ODATA_SRV		<input type="checkbox"/>		Co-deployed only
BEP	/IWEND/GWDEMO_SP2	1 ZCL_ZTEST_GWDEMO_DPC_EXT	GWDEMO_SP2	/WBEP/	<input type="checkbox"/>		Co-deployed only
BEP	ZGWSAMPLE_BASIC	1 SAP Gateway Sample Service - Basic	GWSAMPLE_BASIC	/WBEP/	<input type="checkbox"/>		Co-deployed only
BEP	ZHCMFAB_BEN_ENROLLMENT_SRV	1 HCM Benefit Enrollment	HCMFAB_BEN_ENROLLMENT_SRV		<input type="checkbox"/>		Co-deployed only
BEP	ZHCMFAB_COMMON_SRV	1 OData Service for reusable parts	HCMFAB_COMMON_SRV		<input type="checkbox"/>		Co-deployed only
BEP	ZHCMFAB_EMPLOYEELOOKUP_SRV	1 Employee Lookup	HCMFAB_EMPLOYEELOOKUP_SRV		<input type="checkbox"/>		Co-deployed only
BEP	ZHCMFAB_LEAVE_REQUEST_CR_SRV	1 HCM Leave Request Create - FabFiori	HCMFAB_LEAVE_REQUEST_CR_SRV		<input type="checkbox"/>		Co-deployed only
BEP	ZHCMFAB_MY_EXT_ORG_SRV	1 My External Organizations	HCMFAB_MY_EXT_ORG_SRV		<input type="checkbox"/>		Co-deployed only
BEP	ZHCMFAB_MY_MEDICAL_INFO_SRV	1 My Medical Information	HCMFAB_MY_MEDICAL_INFO_SRV		<input type="checkbox"/>		Co-deployed only
BEP	ZHCMFAB_MY_ADDRESSES_SRV	1 HR Fiori Service: IT0006 Addresses	HCMFAB_MYADDRESSES_SRV		<input type="checkbox"/>		Co-deployed only
BEP	ZHCMFAB_MYBANKDETAILS_SRV	1 Bank details	HCMFAB_MYBANKDETAILS_SRV		<input type="checkbox"/>		Co-deployed only

ICF Nodes: A table showing ICF Node status, Session Time-out, Soft State, and Description. One entry is shown: ODATA, 00:00:00, Standard Mode.

Status	ICF Node	Session Time-out	Soft State	Description
OK	ODATA	00:00:00		Standard Mode

System Aliases: A table for managing system aliases. It includes columns for SAP System Alias, Description, Default System, and Metadata.

SAP System Alias	Description	Default System	Metadata

Transaction /WFND/MAINT_SERVICE: Your Control Center

Central Hub Features:

- See every active service
- Check service status
- Make adjustments when needed
- Everything in one place

Your restaurant management dashboard for OData services

The screenshot displays the SAP Fiori interface for the transaction /WFND/MAINT_SERVICE. The top navigation bar shows the title "Activate and Maintain Services" and the path "/WFND/MAINT_SERVICE".

Service Catalog: This section lists various OData services. A table provides details such as Type, Technical Service Name, Service Description, External Service Name, Namespace, OAuth, Soft State Status, and Processing Mode.

Type	Technical Service Name	Service Description	External Service Name	Namespace	OAuth	Soft State Status	Processing Mode
BEP	GFD_CONFIG_SRV	1 Generic Form Designer Configuration details	GFD_CONFIG_SRV		<input checked="" type="checkbox"/>	Not Supported	Co-deployed only
BEP	GHO_ODATA_SRV	1 OData Service for Global Hydrocarbon Operations	GHO_ODATA_SRV		<input type="checkbox"/>		Co-deployed only
BEP	/WFND/GWDEMO_SP2	1 ZCL_ZTEST_GWDEMO_DPC_EXT	GWDEMO_SP2	/WBEP/	<input type="checkbox"/>		Co-deployed only
BEP	ZGWSAMPLE_BASIC	1 SAP Gateway Sample Service - Basic	ZGWSAMPLE_BASIC	/WBEP/	<input type="checkbox"/>		Co-deployed only
BEP	ZHCMFAB_BEN_ENROLLMENT_SRV	1 HCM Benefits Enrollment	HCMFAB_BEN_ENROLLMENT_SRV		<input type="checkbox"/>		Co-deployed only
BEP	ZHCMFAB_COMMON_SRV	1 OData Service for reusable parts	HCMFAB_COMMON_SRV		<input type="checkbox"/>		Co-deployed only
BEP	ZHCMFAB_EMPLOYEELOOKUP_SRV	1 Employee Lookup	HCMFAB_EMPLOYEELOOKUP_SRV		<input type="checkbox"/>		Co-deployed only
BEP	ZHCMFAB_LEAVE_REQUEST_CR_SRV	1 HCM Leave Request Create - FabFiori	HCMFAB_LEAVE_REQUEST_CR_SRV		<input type="checkbox"/>		Co-deployed only
BEP	ZHCMFAB_MY_EXT_ORG_SRV	1 My External Organizations	HCMFAB_MY_EXT_ORG_SRV		<input type="checkbox"/>		Co-deployed only
BEP	ZHCMFAB_MY_MEDICAL_INFO_SRV	1 My Medical Information	HCMFAB_MY_MEDICAL_INFO_SRV		<input type="checkbox"/>		Co-deployed only
BEP	ZHOMFAB_MYADRESSES_SRV	1 HR Fiori Service: IT0006 Addresses	HCMFAB_MYADRESSES_SRV		<input type="checkbox"/>		Co-deployed only
BEP	ZHCMFAB_MYBANKDETAILS_SRV	1 Bank details	HCMFAB_MYBANKDETAILS_SRV		<input type="checkbox"/>		Co-deployed only

ICF Nodes: This section shows the configuration of ICF nodes, including Status, ICF Node, Session Time-out, Soft State, and Description.

Status	ICF Node	Session Time-out	Soft State	Description
OK	ODATA	00:00:00		Standard Mode

System Aliases: This section allows managing system aliases, including Add System Alias, Remove System Alias, Customizing, and Service Implementation.

Three Operational Areas



Service Catalog

Complete inventory view



ICF Nodes

Communication foundation layer



System Aliases

Backend connectivity mapping

Service Management Sections:

- Service Catalog - Complete inventory view
- ICF Nodes - Communication foundation layer
- System Aliases - Backend connectivity mapping

Each area handles specific management functions, like different kitchen sections

Step One: Access the Service Catalog

01

-  Enter transaction /IWFND/MAINT_SERVICE

03

-  Complete inventory of registered services

02

-  View Service Catalog section

04

-  Single view shows everything

Your Mission Control Dashboard:

Think of it as having a complete inventory system for your OData landscape

[VISUAL SUGGESTION: SAP screen showing Service Catalog section highlighted with yellow border, list of OData services visible with status indicators]

ICF Nodes: Your Diagnostics Center

Communication Foundation Layer:

-  Internet Communication Framework services
 -  Test individual ICF services
 -  Verify services are active
 -  Diagnostics for connectivity issues
- ICF handles HTTP layer; OData handles business logic - both necessary like internet + browser

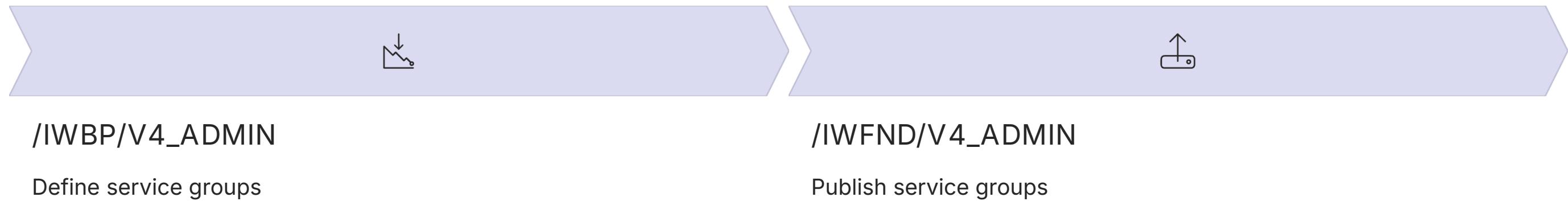
System Aliases: Backend Connectivity Map

Eliminate Troubleshooting Guesswork:

-  Shows which backend systems services connect to
-  Processing mode "deployed only" = no remote connectivity needed
-  Multiple S/4HANA/ECC systems clearly listed
-  Essential for 10+ connected system landscapes

This shows exactly which backend systems your services connect to

OData Version 4: Two-Transaction Approach

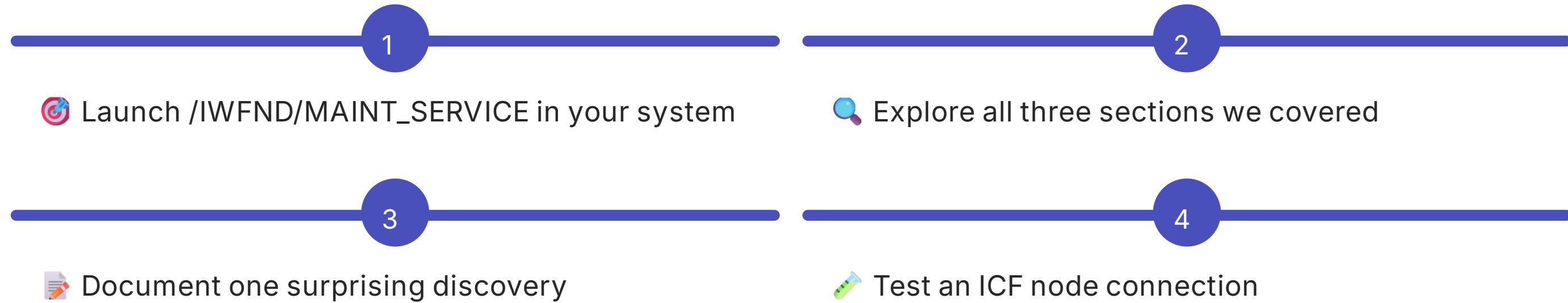


Split Management Process:

- 🎯 /IWB/V4_ADMIN - Define service groups
- 📤 /IWFND/V4_ADMIN - Publish service groups
- 🔍 Use separate sessions for each transaction
- 🌐 S/4HANA Cloud: Fiori notification service appears first

Same management principles apply with cloud-specific configurations

Your 24-Hour Action Plan



Hands-On Discovery Mission:

What surprised you most about your environment's setup?