aSaaS

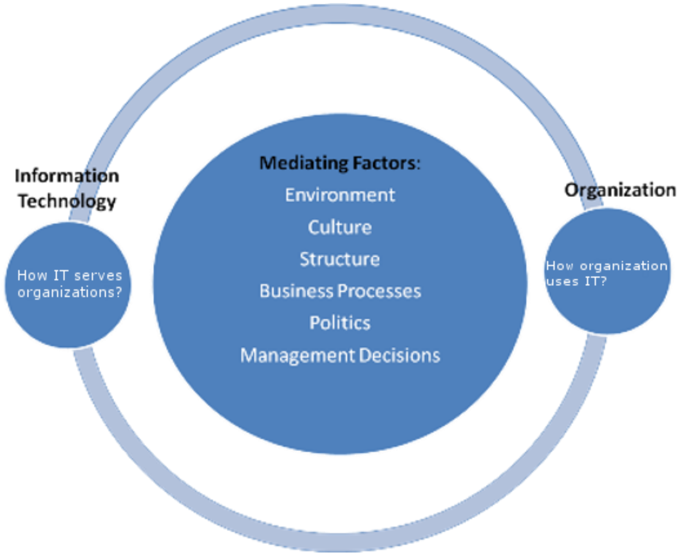
Ext.prd.002.03 | 05.10.2025
UPM 25.x

Table Of Contents

aSaaS

ZINFI's adaptive software-as-a-service (aSaaS) channel management platform:

- Delivers value by helping customers enhance productivity, reduce operating costs and improve processes that build a high-performing channel.
- Enables end-to-end channel management by allowing organizations to continually adapt their channel management infrastructure to their evolving business needs.
- Accelerates your time to deployment, and can be easily configured as requirements change. It provides a highly adaptive and flexible framework for collaboration, and its customizable reporting yields insights that can improve your revenue plans, bookings forecasts, channel programs and partner performance.



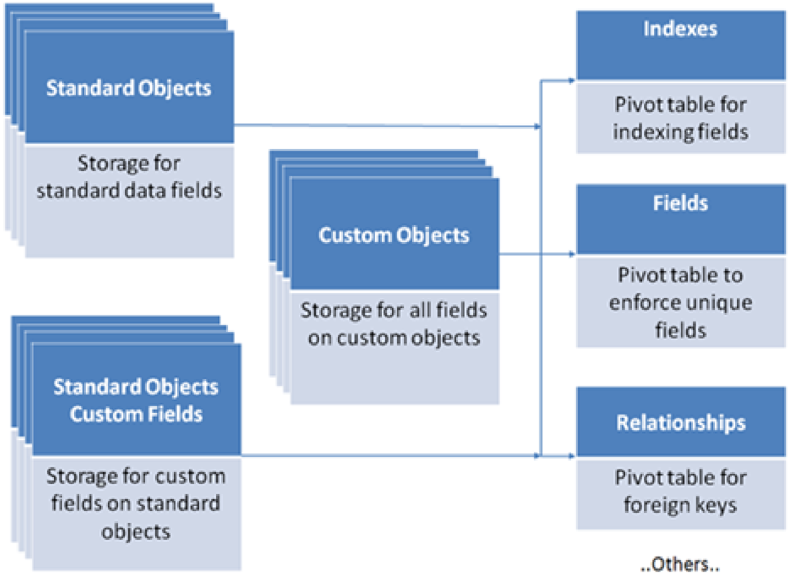
Adapt Your System to its Business Environment

Manage your business, your way. No two small businesses are identical. Configure your system to match your business processes. Our method screens/form layouts are not designed by software programmers! Instead, they are designed by people who use the same drag-and-drop tools that are available to you. The possibilities are endless…customize your ZINFI UPM to match your way of getting your work done, 100% code-free!

aSaaS

Reduce Processing Time

The platform helps you compete and win against the largest competitors. The technology is sophisticated, but the result is simplicity itself and it is affordable because it leverages state-of-the-art technologies like MVVM architecture with a framework dedicated to AngularJS and .Net Core 2.0. With aSaaS, you can build a custom application to address the needs of your business or industry niche in a matter of weeks, without writing a line of code.


Standard Objects
Storage for standard data fields
Custom Objects
Storage for all fields on custom objects
Standard Objects Custom Fields
Storage for custom fields on standard objects
Indexes
Pivot table for indexing fields
Fields
Pivot table to enforce unique fields
Relationships
Pivot table for foreign keys
..Others..

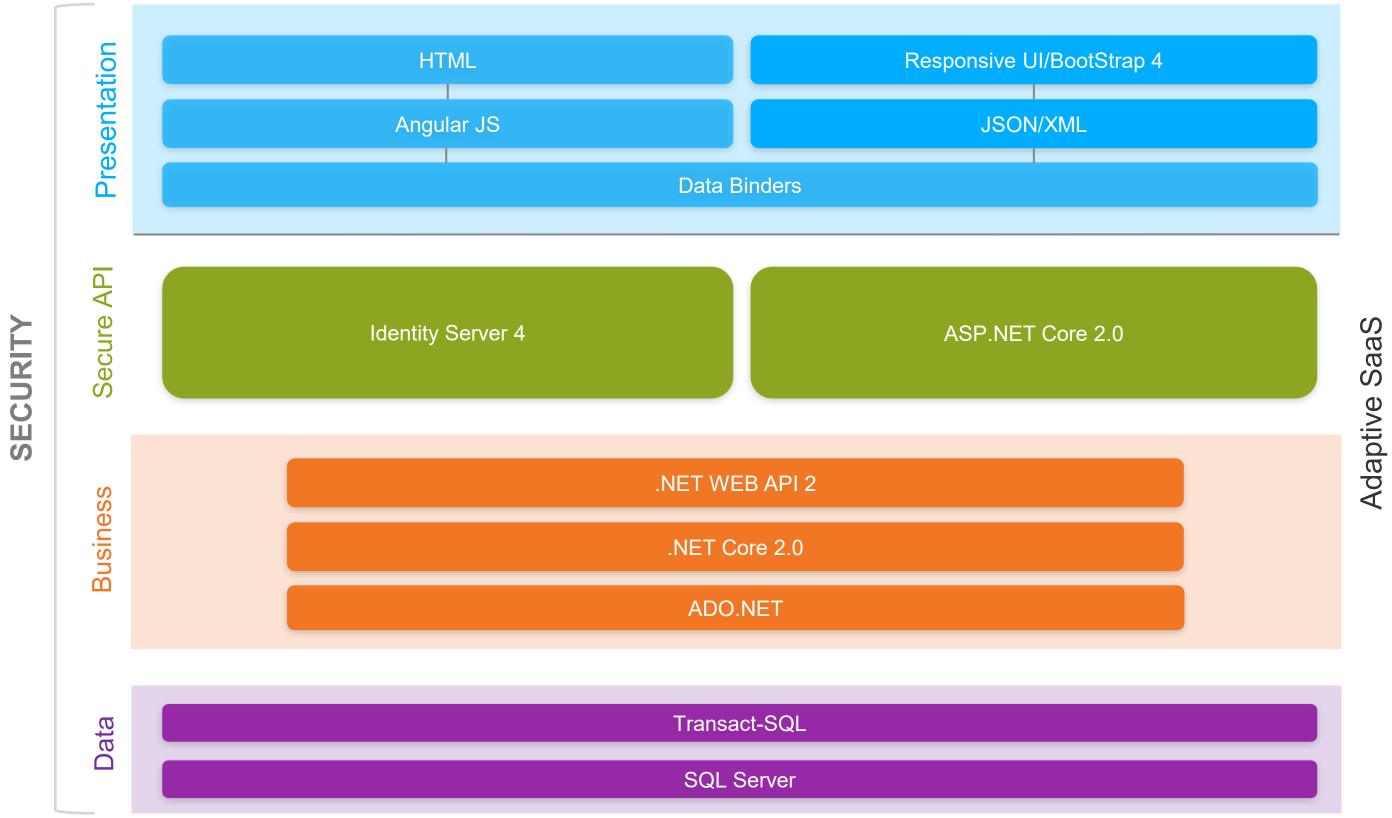
Custom Fields & Tables

These aren't the ho-hum “custom fields” that you see in traditional programs! These are true database fields that you create and use everywhere in method, just like the pre-built fields. Field types can be text, dates, numbers, currencies or file attachments. With aSaaS, you can customize database tables and dropdown list fields.

Custom Buttons, Actions and Validations

Behind every button is a list of actions that run when it is clicked. These actions are set up using a wizard and can lead to some pretty amazing things, like custom-validating a lead, adding a module or looping through a list of terminologies. Actions are the heart of aSaaS, and they replace the need for expensive and complex code or scripting.

Architecture: Technical



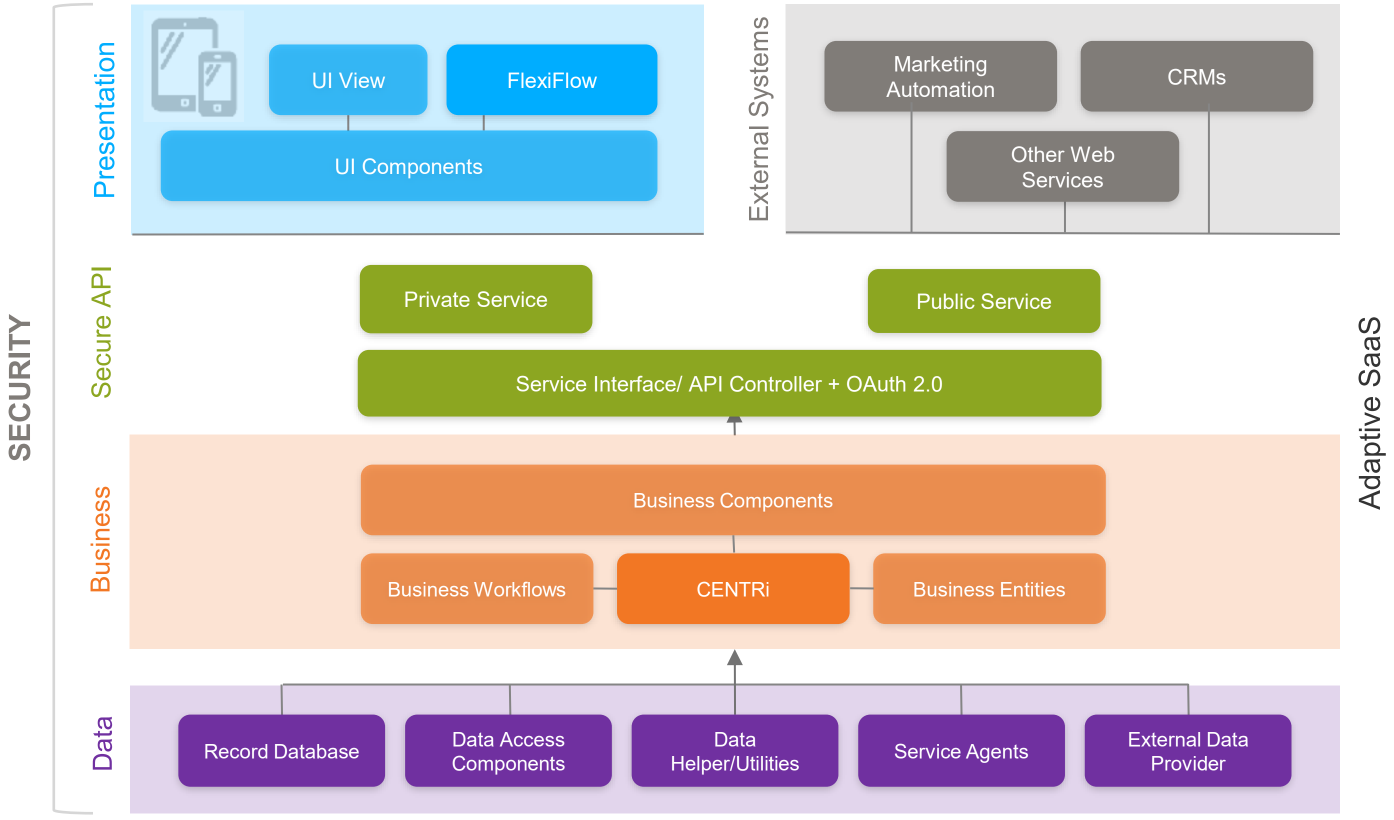
Adapt Your System to its Business Environment

- Custom fields and tables
- Custom buttons, WPF/OLE objects
- Custom actions and validations
- Create custom reports, charts, and graphs

Adaptation in Moments

- Build an enterprise-class custom application in a couple of weeks, without coding
- Scale to address multiple partners/customers
- Upgrade painlessly to receive program enhancements with no impact on your customizations

Architecture: Functional



Completely Mobile:
Architecture translation to both desktop and mobile displays

Efficiently Intuitive:
Desktop app-like user experience which is faster and utilizes less bandwidth

More Secure:
Unified web and native logins; different websites can access portal resources

Highly Flexible:
No dependency between the layers

Fully Modular:
More modularity in different layers allows easy integration with existing installation

Infrastructural Advantages

ZINFI's aSaaS platform gives your channel management organization the ability to build your channel your way. No two businesses are identical. That's why you need a platform that can be rapidly configured to match your unique channel management processes. The possibilities are endless! You can customize your ZINFI platform to match how you get your work done, 100% code-free.

Our mobile architecture translation empowers your organization with both desktop and mobile displays. With no dependency between modules, the platform can easily morph into your business requirements, allowing easy integration with existing installations.

ZINFI's aSaaS application architecture can be easily configured and is fast, mobile-responsive, and native application-friendly. The platform is developed with state-of-the-art web technologies.

This architecture also uses a component-based development process using a SPA (single page application) framework. SPA allows the data fields to refresh instantly without reloading the entire page. This fast refresh capability makes the web access experience equivalent to a desktop app experience. The user interface responsively supports different screen sizes–desktop, tablet, mobile etc.—which makes navigation quite easy.

Key features include:

- Enterprise-class application configuration
- Application scalability management
- Painless application upgrade management

UI Component

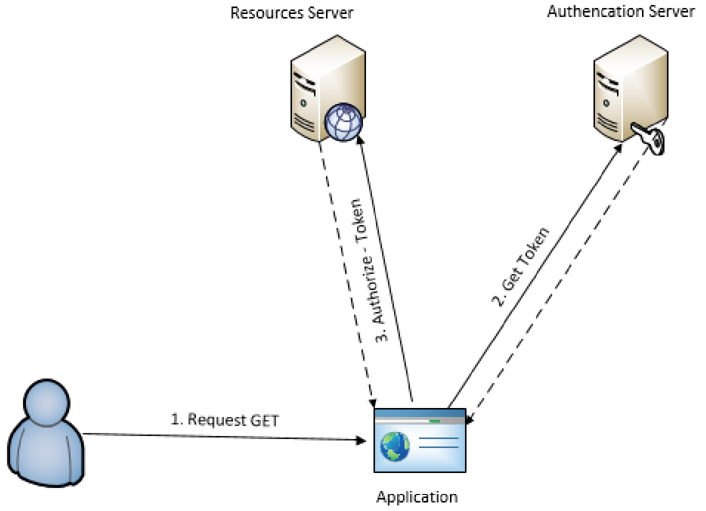
Technology used: AngularJS 5, Bootstrap 4

ZINFI implements component-based development architecture for UPM development for better quality, accelerated development, and reduced risk. The advantages are:

- **Improved application design architecture:** It is very easy (even for new developers) to understand what the application does, and the underlying code logic is easily traceable.
- **Promotes code reusability:** The development of most of the components has been re-architected, empowering the re-use of code in multiple applications without changing a single line of code.
- **Plug-and-play components:** To integrate one of our existing components into a new application, copy it to the components directory, and you're done! It works like magic! All scripts, stylesheets, and static assets are available automatically.
- **Easy to remove components:** It's easy to remove any component. Just delete the directory, and all static assets and stylesheets are removed from the application.
- **Allows better teamwork:** Because of the fact that each component is isolated in its own directory, team members can work on the same project concurrently and with increased efficiently. There are a lot less merge conflicts and overall project progress becomes much easier to track and manage.
- **Consistency: Consistent "look and feel"** enhances ease of use—users will become familiar with how one product functions (looks, reads, etc.) and can translate their experience to other products with the same look and feel.

Authentication Server

- Technology used: Identity Server 4.0, ASP.NET Core 2.0, OAuth 2.0
- An authentication server is used to protect different APIs available in UPM. Identity Server 4.0 provides the OpenID and OAuth 2.0 services. ASP.NET Core is used for creating the UI interfaces for login/logout. ZINFI used OpenID Connect Implicit and Hybrid flow depending on the requirements.



Advantages of using separate authentication server:

- De-couples the authentication logic from the product.
- Allows development of new module as add-on basis, enabling rapid development and release.
- Provides granular control over different private and public client applications.
- Client-specific resource access grants.
- Overall, provides better-secured SSO with other services.

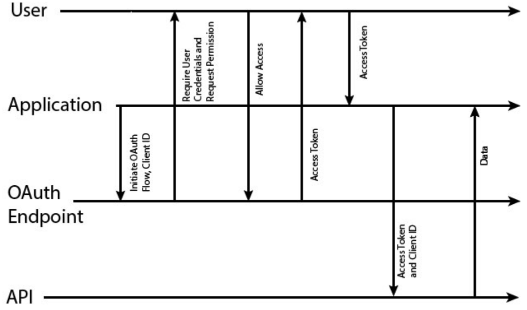
OAuth 2.0

In OAuth 2.0, the following three basic components are involved:

- **The user**, who possesses data that is accessed through the API and wants to allow the application to access it
- **The application**, which accesses the data through the API on the user's behalf
- **The API**, which controls and enables access to the user's data

Using OAuth 2.0, it is possible for the application to access the user's data without the disclosure of the user's credentials to the application. The API will grant access only when it receives a valid access token from the application. How the application obtains an access token is dependent upon the OAuth scheme that is in use.

A public scheme is suitable when an application is incapable of maintaining the secrecy of what OAuth calls “the client_secret.” This is usually the case when the application is native on a computer or mobile where the secret would have to be stored on the user's device, likely inside the source code of the application. As such, these schemes do not make use of the client secret. The ZINFI platform is uses this flow.


User
Application
OAuth Endpoint
API
Initiate OAuth Flow, Client ID
Require User Credentials and Request Permission
Allow Access
Access Token
Access Token
Access Token and Client ID
Data

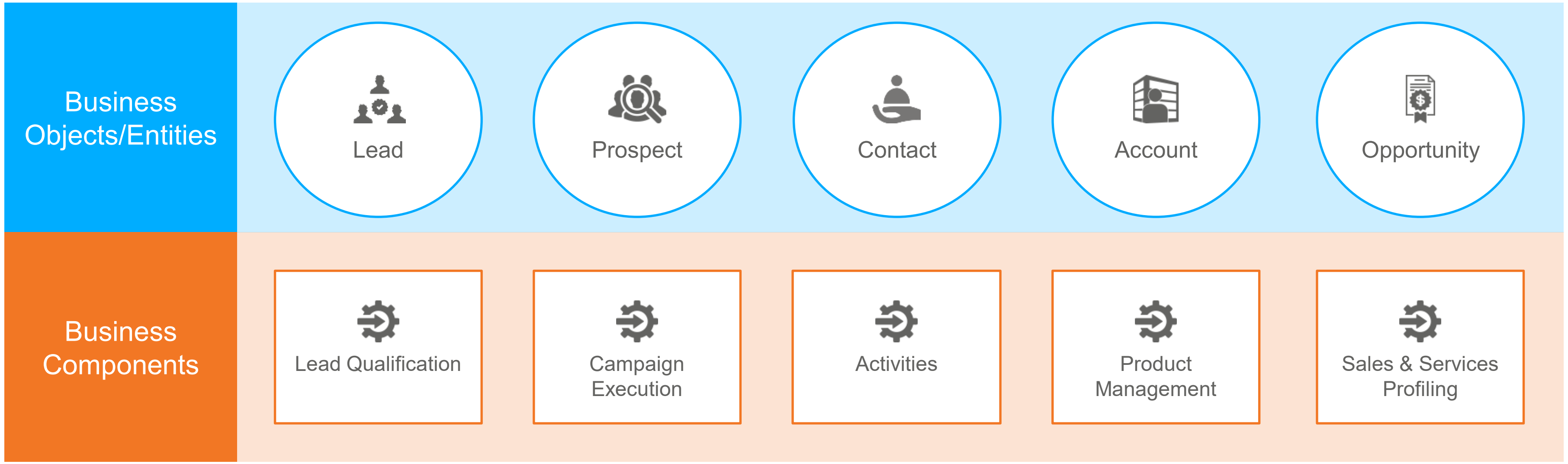
Implicit Flow

In the implicit flow scheme, the application requests an access token from the gateway server and the user grants permission, at which point an access token is provided to the user, who must then pass the token to the application

Business Components and Entities

Business components are automated via the following procedures:

- **People and objects:** Creation of objects and entities
- **Processes:** Workflows, code logic automated through code optimization and stored procedures, CENTRi
- **Technology:** MVVM (Model, View, View Model) paradigm implemented through AngularJS and .NET Core 2.0



Business Workflow

Business Workflows is a modern line-of-business case management solution that:

- Can define and automate workflows, and manage cases with ease and efficiency
- Uses defined business rules to route work to the right system or the right person, and to prioritize work within queues
- Automates task execution, capturing needed information to display detailed insights at just the right time

Business Workflows are integrated through the following state-of-the-art technologies:

- Data models – Object Mappings, Table Architecture, Data Definition/Data Manipulation Logic
- Business rules – Lead Conversion Logic, Partner Acquiring Process, Marketing Management, MDF
- Business workflows – Partner Communication/Onboarding/Business Planning, Lead Management, Communication Management, Product Management, Marketing Process
- User interface templates – Lead Summary, Campaign Execution Summary, Conversion Summary

Main advantages:

- Users can quickly convert new business goals into new business workflows and processes with minimal reliance on IT staff
- Automates the programming required to turn business rules and workflow models into running applications, greatly accelerating your rollout of new processes and services
- Business rules are sophisticated and flexible, and powered by a high-performance business rules engine, enabling you to fully capture your organization’s objectives and insights into rules-driven enterprise applications

Record Database

ZINFI uses SQL Server 2016 on the back end to manage our databases.

With SQL Server 2016, we can build intelligent, mission-critical applications using a scalable, hybrid database platform that has everything built in, from in-memory performance and advanced security to in-database analytics.

There are several unique advantages of using SQL Server 2016 as a database engine:

- New Query Store stores query texts, execution plans and performance metrics within the database, allowing **easy monitoring** and troubleshooting of performance issues.
- Temporal tables are **history tables**, which record all data changes, complete with the date and time they occurred.
- Multiple tempDB database files can be configured during SQL Server installation and setup.
- New built-in JSON support in SQL Server supports JSON imports, exports, parsing and storing.
- New PolyBase query engine integrates SQL Server with external data in Hadoop or Azure Blob storage. We can import and export data as well as execute queries.
- New **security** features:
    - Always encrypted: When enabled, only the application that has the encryption key can access the encrypted sensitive data in the SQL Server 2016 database. The key is never passed to SQL Server.
    - Dynamic data masking: If specified in the table definition, masked data is hidden from most users, and only users with UNMASK permission can see the complete data.
    - Row-level security: Data access can be restricted at the database engine level, so only relevant data can be seen.

Data Helper

SQL Helper Class is used in Data Access Layer, which interacts with the database with the help of the connection string provided and contains several methods (see below). It also improves the performance of the Business Layer & Data Access Layer. We use the following in it:

- ExecuteNonQuery: Returns the number of rows affected which is an INTEGER value.
- ExecuteDataset: Returns the DataSet object, which can contain single or multiple tables containing records and can be used to populate controls like GridView, Repeater, etc.
- ExecuteDataTable: Returns a table using an SQL query on the back end.
- ExecuteReader: Returns the SQLDataReader object, which can be used in loop for reading records one by one, and can also be directly assigned to a DataSource control.
- ExcuteScalar: Returns the object of the cell value selected in the SQL query.

ZINFI developed a custom data access wrapper class using these standard SQL Server helper methods so the application developers do not need to know details about the database connectivity and can access data using the customized methods. Example: Remove, RemoveAt, Insert, InsertAt etc.

- Provides the ability to update the technology stack in another tier without impacting the data access tier.
- Provides ease of maintenance of the code base
- Enhances the security of the application.

Data Access Components

The Microsoft® Data Access Components (MDAC) are the key technologies that enable Universal Data Access. Data-driven client/server applications deployed over the web or a LAN can use these components to easily integrate information from a variety of sources, both relational (SQL) and nonrelational. These components include the following:

- **ActiveX Data Objects (ADO):** Microsoft ActiveX Data Objects (ADO) is the strategic application programming interface (API) to data and information. ADO provides consistent, high-performance access to data and supports a variety of development needs, including the creation of front-end database clients and middle-tier business objects that use applications, tools, languages or Internet browsers. ADO provides an easy-to-use interface to OLE DB, which provides the underlying access to data. ADO is implemented with minimal network traffic in key scenarios, and a minimal number of layers between the front end and data store—all to provide a lightweight, high-performance interface.

- **OLE DB:** OLE DB is the Microsoft strategic system-level programming interface to data across the organization. OLE DB is an open specification designed to build on the success of ODBC by providing an open standard for accessing all kinds of data. Whereas ODBC was created to access relational databases, OLE DB is designed for relational and nonrelational information sources, including mainframe ISAM/VSAM and hierarchical databases; e-mail and file system stores; text, graphical, and geographical data; custom business objects; and more. OLE DB components consist of data providers, which contain and expose data; data consumers, which use data; and service components, which process and transport data (such as query processors and cursor engines)

- **Open Database Connectivity (ODBC):** The Microsoft Open Database Connectivity (ODBC) interface is an industry standard and a component of Microsoft® Windows® Open Services Architecture (WOSA). The ODBC interface makes it possible for applications to access data from a variety of database management systems (DBMSs). ODBC permits maximum interoperability—an application can access data in diverse DBMSs through a single interface. Furthermore, that application will be independent of any DBMS from which it accesses data. Users of the application can add software components called drivers, which create an interface between an application and a specific DBMS.

External Data Provider

The External Data Provider enables integration of external data into the ZINFI database engine. It can access data from a variety of external data sources—like CRM platforms, Salesforce, MS Dynamics, Google AdWords, Dropbox, etc.—through API services.

The external data is processed by a centralized engine, ZINFI's CENTRi/Connectors

CENTRi/Connectors helps set up connections with external systems easily without compromising security by developing a completely different custom connectivity engine.

Data Layer Features

SQL Server 2016

- Easy monitoring & troubleshooting
- Temporal tables automate historical records
- Import & export data while executing queries
- Always encrypted
- Dynamic data masking
- Row-level security

SQL Data Helper

- Improves business layer & data access layer performance
- Allows updating of technology stack without impacting data access
- Code base “Ease of Maintenance”
- Enhances application security

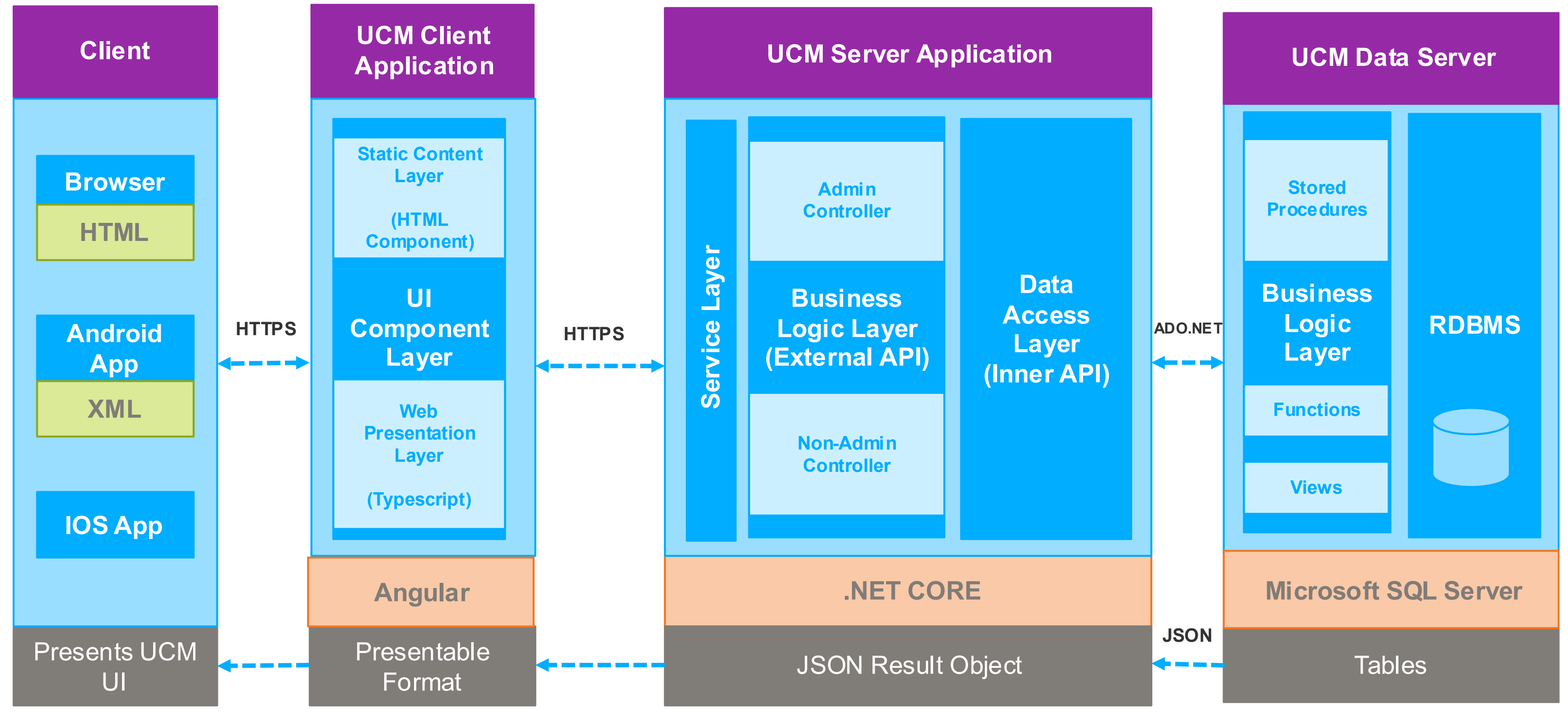
External Data Provider

- Enables integration of external data into the ZINFI database engine
- Accesses a variety of external sources through APIs
- Easy connection setup without compromising security

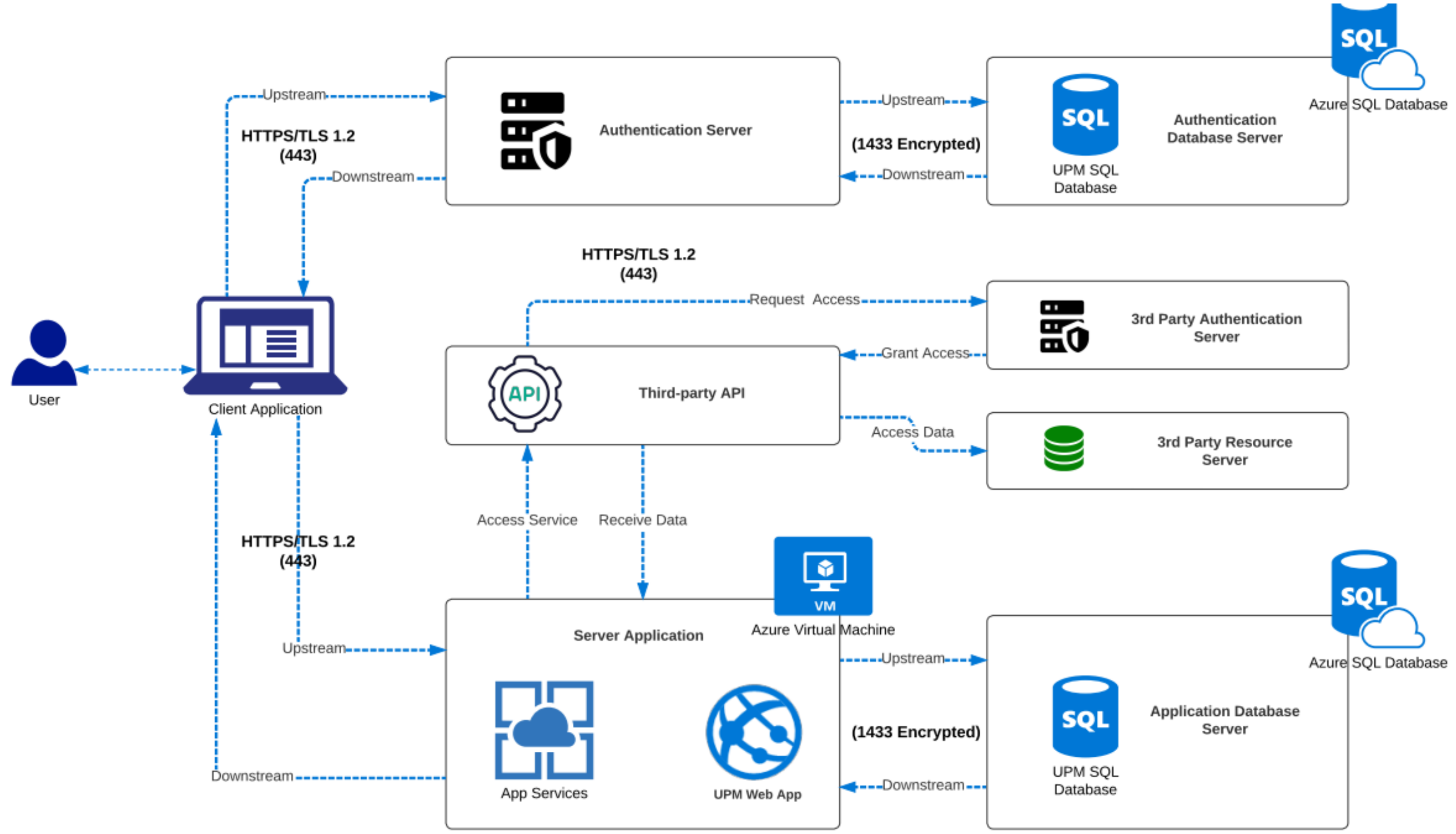
Data Access Components

- These components easily integrate information from a variety of sources:
  - **ActiveX Data Objects (ADO)**
  - **OLE DB**
  - **Open Database Connectivity (ODBC)**

Futuristic Architecture (implemented through aSaaS)



The Authentication and Third-Party Connector System



Summary

Client Application

This is the UI of application, which communicates with the **Server Application** through HTTPS. The application UI is available in two formats:

Desktop – The application runs through browser.

Mobile

a. **Android** – The application runs through a native app exclusively made for Android.

b. **iOS** – The application runs through a native app made exclusively for iOS.

Authentication Server

Used for authenticating the client log-in request, in one of two ways:

- By retrieving the application token from **Authentication Data Server**, when logged in directly using ZINFI UPM.
- By retrieving the application token from **Authentication Data Server** via a third-party API, when logged in using CRM login credentials utilizing single sign-on (SSO) functionality; for example, Salesforce, MS Dynamics etc.

Third Party App

Used for accessing third-party data to achieve SSO functionality.

Authentication Data Server

Used for providing the application token to the authentication server.

Summary

Server Application

This section consists of the UI layer logic and component layer logic to interact with the data layer for information retrieval. It consists of the following layers:

- **UI Component Layer** – This section is designed in AngularJS using TypeScript.
  - **Static Content Layer** – This layer contains the HTML components.
  - **Web Presentation Layer** – This layer contains the TypeScript code files, which get embedded with HTML components to produce the desired design of the page.
- **Service Layer** – This section is designed in AngularJS using TypeScript.
  - **Internal / MVVM Controller** – This layer consists of the handler and service component classes written in TypeScript. These classes are then invoked in Web Presentation Layer to get the required outcome.
- **Logic Layer [External API]** – This section is designed in .NET Core 2.0. When the control enters from AngularJS to .NET Core, it first hits the Logic Layer.
  - **System Controller** – This layer contains the API classes and method calls for Administrative module(s) of ZINFI UPM.
  - **Values Controller** – This layer contains the API classes and method calls for rest of the modules of ZINFI UPM, except the administrative module(s).
- **Data Access Layer [Inner API]** – This section is designed in .NET Core 2.0. This layer contains the method definitions that are called from the Logic Layer. With the help of ADO.NET Technology, ZINFI UPM communicates with Data Server for retrieval of data (as required).

Summary

Data Server

This section contains the business logic of ZINFI UPM in **Business Logic Layer** and **Relational Database** to store application data. Microsoft SQL Server is used for this purpose.

- **Business Logic Layer** – Contains Stored Procedures, Functions and Views which constitute the business logic of ZINFI UPM.
- **Relational Database** – Contains raw data along with their relational constraints.

OpenID-OAuth Login Process

Authentication Server Validation Checks

The new architecture follows the **OpenID Implicit Flow** methodology. In this new process, when a user logs into the application, the authentication process follows Implicit Flow steps:

1. Client Application further referred to as Client (JavaScript / C#) prepares an Authentication Request containing the desired request parameters.
2. Client sends the request to the Authentication Server.
3. Authentication Server authenticates the End-User.
4. Authentication Server obtains End-User Consent/Authorization.
5. Authentication Server sends the End-User back to the Client with an ID Token and, if requested, an Access Token.
6. Client validates the tokens and retrieves the End-User's Subject Identifier.

ZINFI Technologies, Inc.
6200 Stoneridge Mall Road, Suite 300
Pleasanton, CA 94588

sales@zinfitech.com