# finPOWER Connect
# Architecture Guide

Version 1.01
19 August 2024

# Contents

# Disclaimer

This document is for informational purposes only. All information contained in this document is provided "as is" without warranty of any kind, and Intersoft accepts no liability for any decisions made on the basis of this information This document contains information that may be subject to change at any stage.

# Version History

| Date | Version | Name | Changes |
|---|---|---|---|
| 10/07/2024 | 1.00 | JR | Created |
| **19/08/2024** | **1.01** | **JR/PH** | **Updated to include additional information regarding version 6** |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

# Introduction

finPOWER Connect is designed around the concept of a core set of Business Layer functions with a flexible scripting system built in to allow customisation.

This architecture means the system can be customised to handle many forms of lending and funding.

A Multi-Tiered architecture ensures a modular and scalable design, allowing for easy adaptation to evolving technologies and requirements.

finPOWER Connect currently uses Microsoft .NET Framework 4.8. In 2024/ 2025 we will be moving to .NET 8 with a release scheduled for early 2025.

finPOWER Connect is designed to run either as a Windows Desktop application or Web Application running under ASP.NET, or both. finPOWER Connect version 6 will use ASP.NET Core.

Customers host their own Database, Application and Web Server, albeit potentially using a hosting platform such as Microsoft Azure or Amazon Web Services (AWS). In addition, Intersoft can provide hosting in either a shared or dedicated environment, depending on a customer's needs, via the Microsoft Azure hosting platform – thus providing a simple SaaS environment for the customer.

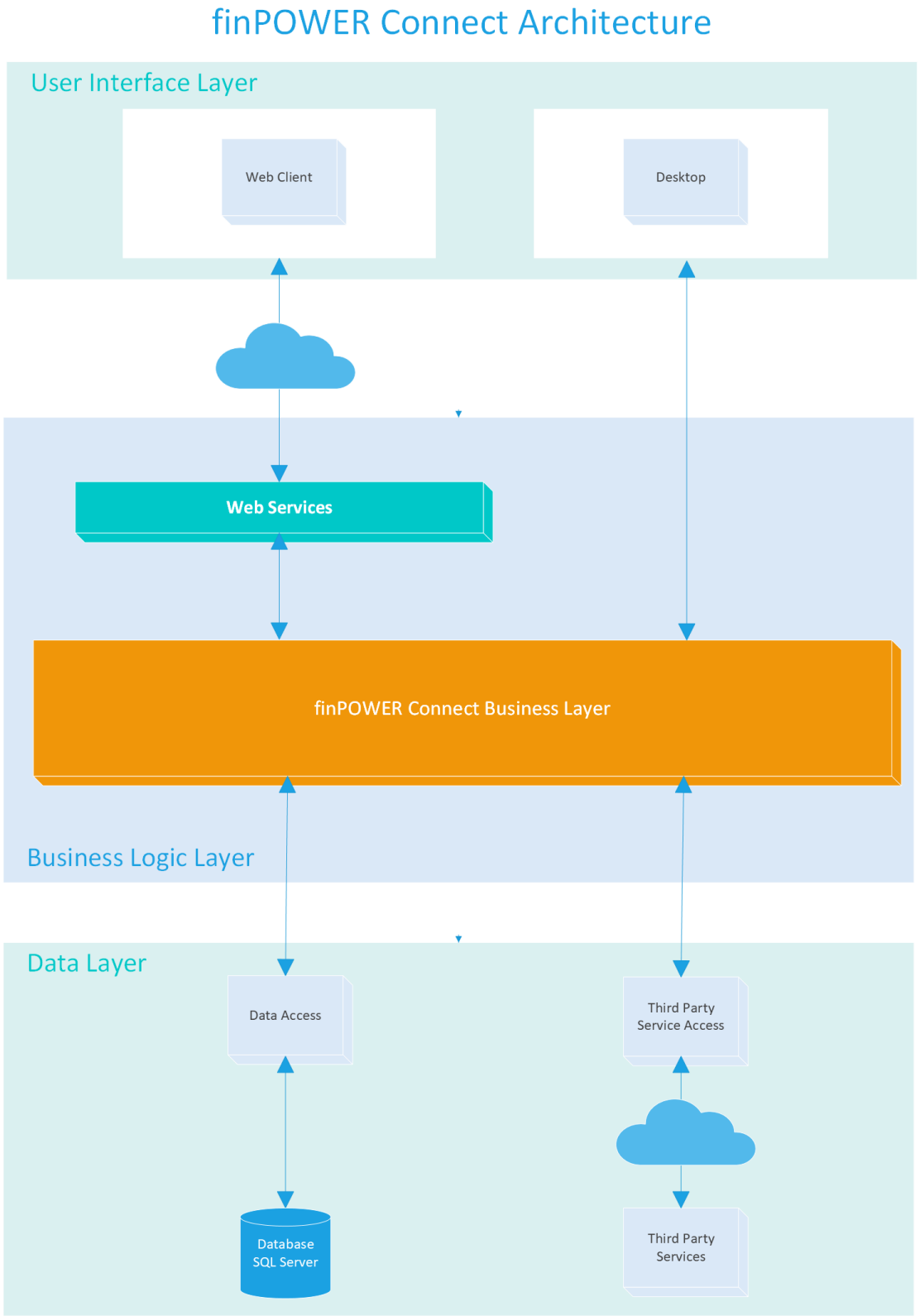
# Architecture

finPOWER Connect is a multi-tiered layer application.

Multi-tier architecture is a well-established software application architecture that organizes applications into three or more logical tiers.

This includes the Presentation tier, or user interface; the Business Logic tier, where data is processed; and the Data tier, where the data associated with the application is stored and managed.

This allows separation between the User Interface and Business Logic, providing benefits such as scalability, security and code maintainability.



# User Interface (Presentation) Layer

## Windows Forms UI

This was our original UI for the Windows desktop.

## Web/ Cloud based UI

Our more recent development is moving our traditional Windows forms UI to a web-based UI.

Almost all forms are customisable using what we term "HTML Widgets" – see later. This allows great flexibility and customisability for Customers.

## Web Portal UI and development tools

We can develop customised Web Portals for User sites, for such things as Client Portals (e.g. Customer Banking Applications) and Dealer Portals etc.

They are built using tools within the finPOWER Connect application itself.

# Business Logic Layer

The core application logic layer that every action must be processed through – it is the heart of the finPOWER Connect application.

- Within this layer, there are various sub-layers such as our Query and Report generator.
- The Presentation Layer does not directly interact with the Data Layer.
  - All functions MUST go through the Business Logic layer.

## Third Party Interface Layer

Whilst this could be considered part of the Business or Data Layers, we separate the underlying API interface elements into separate assemblies.

There are various sub-parts, including:

- Accounting Interfaces
- Banking Interfaces
- Credit Bureaus
- Electronic Signature services
- Messaging services
- Security Enquiry
- Security Registry
  - PPSR in Australia and New Zealand

Business Layer functions requiring access to third party APIs will call this layer.

Higher level functions within the Business layer create requests and examine responses and update data as necessary.

# Data Layer

In finPOWER Connect the Data Layer is abstracted away from the actual Database Engine.

This means we can support different Database Engines without changing the core code, as the technical implementation is a "black box" to the code using it.

In reality, the Data Layer consists of two sub-layers, the finPOWER Connect abstracted black box code and the actual underlying Database Engine.

The underlying Database Engine currently supports:

- Microsoft SQL Server
- Microsoft Azure SQL Database
- Microsoft Access
  - For smaller clients running on a local network.

## Hosted vs Multi-Tenant Cloud-based Application

We do not think a multi-tenanted cloud-based application, like MYOB or Xero, will work for most Finance Companies.

A Finance Company's operation revolves around finPOWER Connect; it is the key application for their business, with very sensitive data. Therefore, in our experience they want control, including separation, over their data.

We would also lose much of our customisation functionality, e.g., scripting, because it would be very difficult to allow this in such an environment, i.e., where one customer could literally bring down everyone else.

Hosted means the Customer is in full control of their own database, no-one else can see their data, including files stored in the Document Manager, and can be hosted as the Customer wishes.

# Scripting Engine

A very important part of finPOWER Connect is its scripting engine.

The scripting system within finPOWER Connect provides both the ability to modify the User Interface (on both the Windows desktop and on the Cloud) and Business Logic layer.

## User Interface

Forms generally consist of fixed set of pages and summary pages (rendered HTML).

Summary Pages are completely customisable, meaning they can reflect the information the customer wants to see and can be dynamically created based upon the type of Account, its status etc.

## HTML Widgets

HTML Widgets allow fully customisable User Interface forms to be created. They can be used to override certain forms within finPOWER Connect and finPOWER Connect Cloud to better suit business requirements.

They use scripting and HTML and JavaScript to provide complete control over the User Interface and, of course, to access the finPOWER Connect Business Logic layer.

The application of HTML Widgets is potentially limitless and can be used wherever there is the need for a custom form or interface within either the Desktop or Cloud UI.

HTML Widgets can be used to override existing forms, e.g. a Loan Application form, or to customise how a Payment Arrangement works.

HTML Widgets can be used to create custom forms, e.g. add specialised forms for Cheque Cashing and CRM forms to drive targeted marketing campaigns.

Note, the Cloud based UI principally uses HTML Widgets for its UI.

Intersoft supplies standard HTML Widgets that are used within finPOWER Connect, meaning that it is easy to take an existing form and customise.

An HTML Widget form can look and feel like any other finPOWER Connect Cloud form and, when running within the desktop version of finPOWER Connect, inherit a similar style to built-in forms.

## Business Logic

By including many events that a script can intercept, finPOWER Connect is extremely customisable, providing a very flexible system.

Of course, scripts cannot override all functionality, for example, whilst a script could stop a Loan from being closed, it cannot close an Loan that does not have a zero balance.

## Portals

Portals represent Web applications that can be designed within finPOWER Connect and then used either within finPOWER Connect Desktop or, more importantly, externally via Web Services.

Examples or what Portals may be used for are:

- Customer Banking Portal
  - E.g., Clients can access the Portal via a Web Browser and view a list of their Accounts.
- Dealer/ Broker/ Agent Portal
  - External Party Users can sign in, add Account Applications and view their Tasks.
- Public Loan Application Portal
  - Typically, a Loan Application form can be implemented as an HTML Widget.

# Tech Stack

The tech stack is the combination of programming languages, frameworks, libraries, tools, and technologies that are used to develop and deploy finPOWER Connect.

Intersoft has always developed software primarily using Microsoft based technologies.

## Technologies

- Windows Operating System
- Database support
  - Microsoft SQL Server
  - Microsoft Azure SQL Database
  - Microsoft Access
- Web Server
  - Microsoft Internet Information Server (IIS)
- On-Premises and hosted Cloud Deployment

## Development Languages and Environment

- Microsoft Visual Studio IDE
  - NuGet package manager
- Microsoft DevOps code repository
- Microsoft .NET Framework
- .NET 8 (from finPOWER Connect version 6)
- VB.NET
- C# (parts of finPOWER Connect version 6)
- ASP.NET
- ASP.NET Core (from finPOWER Connect version 6)
- Microsoft Web API framework
- HTML, CSS and Javascript

In 2024 and 2025 we will be moving from the .NET Framework to .NET. This means some technologies may also change, for example Web Forms to Razor framework.

## Other Tools and Libraries

There are several other tools and libraries that finPOWER Connect utilises.

Note, as far as possible, finPOWER Connect will "black box" these components. This means that externally, and especially within the scripting engine, there is no reference to the underlying component – meaning that the component could be substituted for another in the future.

Some of the major third-party components used in finPOWER Connect include:

- Actipro Syntax Editor
  - The code editor control and parsing suite that is used to code scripts.
  - https://www.actiprosoftware.com/products/controls/windowsform
- ActiveReports
  - The Report generator used.
  - https://developer.mescius.com/activereports
- Gembox Document and Spreadsheet
  - These components enable finPOWER Connect to read and write Microsoft Word and Excel files, as well as creating PDF documents.
  - https://www.gemboxsoftware.com/

- MailKit/ MimeKit
  - An open source Mail library used by finPOWER Connect to send emails, as well as managing mail folders and messages.
  - https://mimekit.net/

## Visual Basic vs C# (C Sharp)

finPOWER Connect is written using Microsoft Visual Basic. We currently have no plans to migrate the core finPOWER Connect Business Layer from Visual Basic to C#.

However, from finPOWER Connect version 6, finPOWER Connect Cloud and Web Services are a mix of VB and C#.

We realise many software programmers prefer C# and therefore provide scripting functionality within finPOWER Connect in either language.

The language syntax between Visual Basic and C# is quite different, however, programming concepts like variables, functions, conditions, loops etc are standard to both languages. Both languages use the same underlying .NET Framework and .NET CLR and were developed together to be part of the same .NET Framework platform.

Finally, there has been speculation that Microsoft is stopping development of Visual Basic, but this is not true. Microsoft offers three languages on the .NET platform – C#, F#, and Visual Basic – and they remain committed to full support for all three languages.

For more information see:

- https://devblogs.microsoft.com/dotnet/update-to-the-dotnet-language-strategy/
- https://learn.microsoft.com/en-us/dotnet/fundamentals/languages