Percipient©
intelligent data engineering

UniConnect Transact™

A Percipient Technology White Paper

Author: Gayathri Dwaraknath
Chief Solutions Officer

Updated Aug 2018

Content

1. Introduction

UniConnect's software platform allows Financial Institution (FI) and Fintech[1] developers to quickly build at-scale Apps and Services and integrate them with a variety of Back-end Services, such as core banking systems, data warehouse systems, payment networks and credit bureaus.

The Fintech promise is to enable the creation of innovative solutions for consumers to access and manage their money and financial assets. Fintech vendors offer the potential to enhance customer experiences by utilizing latest advancements in technology – think chatbots, artificial intelligence, augmented reality, latest generation of smart phones and smart watches. They are able to combine ideas from other fields such as gaming, social media and sharing economy to develop mashups that bring out the best of different worlds together. Fintechs therefore create banking solutions that were never imagined before, with the potential to reach audiences that remain unbanked or under-banked.

By enabling FIs to access, publish and consume data, the UniConnect suite of products help bridge the chasm between Fintechs and FIs, and between modern and legacy technologies.

Vast disruptions on the data technology landscape have also brought about benefits for both FIs and fintechs. These include, to no small degree, the opportunity to optimise costs using Cloud computing, and to manage and store new forms of unstructured data and high volume data using Hadoop technology. The explosion of Open Banking APIs further enhance the prospect of new consumer-friendly apps for comparing product features and prices. Yet many FIs have not yet taken advantage of these developments.

This is because today, developers keen to apply Fintech-enabled and new-tech solutions face insurmountable challenges. Integrating a bank customer's financial data with data from across an individual's data eco-system means navigating the complexities of hybrid architectures, penetrating proprietary banking systems, overcoming impenetrable firewalls, dealing with non-standard messaging, and federating next-gen with past-gen technologies. These challenges only serve to throttle innovation and make the job of new tech adoption extremely hard.

This is where Percipient's products, UniConnect Transact™ and UniConnect Integrate™, come to the rescue. By enabling FIs to access, publish and consume data, the UniConnect suite of products help bridge the chasm between Fintechs and FIs, and between modern and legacy technologies. Through the entire spectrum of small to big data, UniConnect helps FIs turn innovative ideas into workable solutions, and accelerate the deployment of these solutions into a variety of game-changing applications.

This whitepaper focuses on UniConnect Transact™, which is aimed at enriching consumer-facing channels and speed up app development.

[1] Fintech is a portmanteau for financial technology. In this document, it refers to either internal teams or entire companies focused on developing innovative solutions for customers to manage their money and financial assets.

2. Product Goals

Percipient's product executives are all veterans of the Financial Services Industry, with a deep understanding of the challenges and opportunities inherent in the industry's shift towards digitization and automation. This understanding has driven the design principles and product vision behind UniConnect Transact™.Here are the 3 key tenets of our product offering:

2.1. Know your customer

We firmly believe that in order to create a great product, we must first understand our customers and their pain points. Our product roadmap is laser-focused on meeting the needs of our customers, and to make their lives 10x easier. Our customers are business users and developers who seek innovative solutions using the latest generation of digital technologies. We aim to make it remarkably easy for this community to build new age solutions by providing seamless access to data.

Companies and products that are built with efficiency and cost discipline in mind have a better chance to survive. Percipient aim to build software that is affordable to use and operate, so our customers don't have to pay any more than they should.

2.2. Agility and adaptability

The pace of technology evolution in recent times has been phenomenal. For a software platform that fosters innovation, it is paramount to be agile, with the ability to adapt to the constant cycle of technology evolution. This is possible through utilizing new software development methodologies, tools and technologies that fundamentally assume that change is the only constant, and that next year will look very different from this year. With this in mind, UniConnect Transact™ embraces:

- modern agile tools and processes
- highly modular software code and server architecture
- polyglot programming models
- cloud based products from tools to infrastructure
- the latest generation of technologies.

2.3. Be frugal, be wise

Nobody likes to pay more for stuff. Over time, companies and products that are built with efficiency and cost discipline in mind have a better chance to survive. We aim to build software that is affordable to use and operate, so our customers don't have to pay any more than they should. We do this by embracing the following principles:

- Automate as much as possible (human-labor cost savings helps fund computing costs)
- Software code is also the best document, so keep it nice and clean
- Use only as much computing resources as is necessary
- Migrate fixed costs to variable costs wherever possible

3. Architecture Principles

3.1. The Power of APIs

UniConnect Transact™ does all the heavy lifting work of connecting to myriad systems behind FIs' firewalls, and bringing data out in the form of oh-so-easy-to-call APIs.

The large majority of FIs do not have the ability to allow Fintech apps to access their backend systems. This is because most FIs still rely heavily on Mainframe based systems to run their processes, or use software products that have no ability to allow external applications to communicate with them.

UniConnect Transact™ has a comprehensive built-in library of connectors to several well-known core banking systems. Based upon these connectors, we have developed services that allow for a range of inquiry and transaction operations to be performed against backend systems. These services are accessed in the form of modern Web Application Programming Interfaces, or Web APIs, henceforth referred to simply as APIs. The power of APIs lies in the fact that it relies on ubiquitous network protocols and messaging standards such as HTTP(S) and JSON.

A regular web-browser is all that is needed to call an API. APIs are also very easily understood by most software developers, which significantly improves their ability to quickly develop apps that can invoke the APIs and access FI and other data. UniConnect Transact™ does all the heavy lifting work of connecting to myriad systems behind FIs' firewalls, and bringing data out in the form of oh-so-easy-to-call APIs. As such, UniConnect Transact™ enables improved developer outreach, faster product development and accelerated innovation cycles.

3.2. Why we use Open Source Software

Our product is based on a large stack of several high-quality open source software (OSS) products. We believe that OSS currently offers the best choice for latest generation technologies. OSS does not imply free or least-cost. Rather, the effective application of OSS requires highly talented engineers, who are specialists in specific OSS technologies.

As most OSS do not have commercial companies providing support or guarantees, any company which embeds OSS in their products and services must assume the burden of providing reliable support and SLAs. Today, the world's largest tech companies are both generators and users of OSS, thereby rendering meaningless the tradeoff between using OSS and commercially developed products. For Percipient, OSS represents the opportunity to ensure the most advanced offering to FI and Fintech developers.

3.3. Microservices and Service Mesh

In order to achieve the highest degree of agility and adaptability in our product, FIs must leverage the latest trends in developing enterprise software architecture. One such paradigm shift in server-side application development is a Microservices-based architecture.

Simply put, a microservice is a large server-side software that has been broken down into many smaller pieces. Each piece of software is capable of running on its own mini-server, has an independent lifecycle (start, stop, update code, restart), and can be replicated. When required to handle more volume, each microservice can be scaled independent of others.

Quite frequently, one microservice needs to invoke other microservices, and so on. This leads to a mesh of interconnected microservices. UniConnect Transact™ is based on such a service mesh design, consisting of many microservices all operating together as part of a symphony.

3.4. Cloud native design

UniConnect Transact™ fully leverage the scalability, reliability and cost efficiency of present-day Cloud computing technologies.

We have developed UniConnect Transact™ to fully leverage the scalability, reliability and cost efficiency of present-day Cloud computing technologies. Our product has been designed and optimized from ground-up for operating within a Cloud environment, whether that is a public, private or hybrid Cloud infrastructure. We also heavily rely on container technology, based on open source Docker and Kubernetes software. This makes our product very portable, agnostic of underlying Cloud service providers, and thereby achieves very high speed to market.

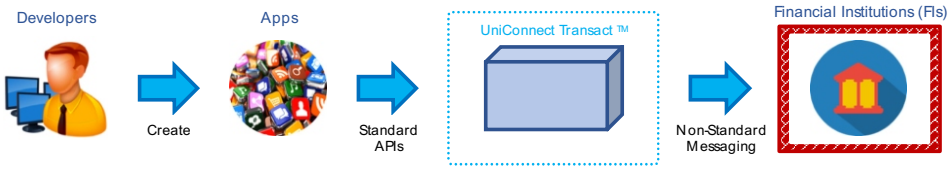
4. High Level Architecture

In this section, we will cover the high-level architecture of UniConnect Transact™.

4.1. Product Overview

The simplest way to understand what UniConnect Transact™ does is to visualize it in context of its primary use-case – allowing Developers to create Apps that can easily integrate with Financial Institutions' transaction systems through a Standard API.

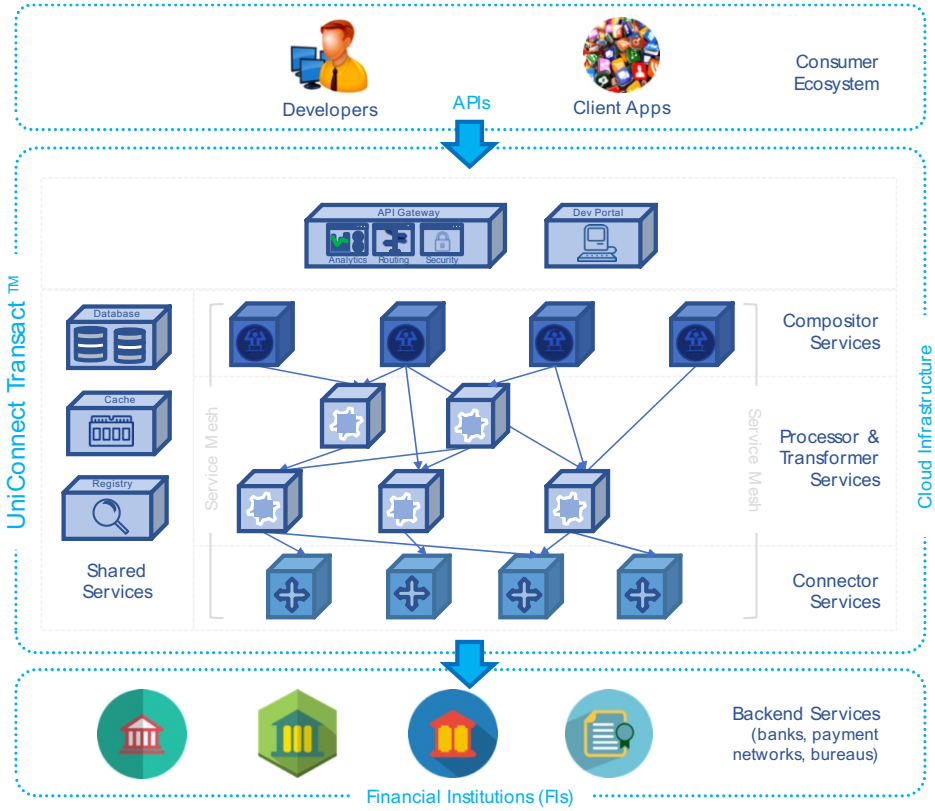
This is depicted in the following diagram:



4.2. Product Details

Now let's take a look inside the UniConnect Transact™ product. Like most modern enterprise software applications, it has a layered architecture.

Here's a detailed description of each of the layers and components in the diagram below:



- **Developers**

  Engineers and product managers who work for Fintechs and produce Apps and Services.

- **Client Apps**

  Software applications and services written to run on the latest generation of mobile phones, smart devices and other digital technologies.

- **API Gateway**

  A network application that is optimized for handling API requests and responses. Among other things, it takes care of request routing, security and access control, traffic monitoring, and producing analytics so businesses can understand their API usage.

- **Dev Portal**

  A website meant for developers to register and get license keys to access APIs. It is also a place to read documentation on the APIs, download sample code, run simple tests against the APIs, and monitor usage of their account.

- **Compositor Services**

  A layer of microservices whose role is to aggregate and orchestrate across several other microservices in order to provide a higher level of functionality. For example, an 'Account List' microservice would need to aggregate responses from multiple other microservices, each specialized in returning accounts for credit cards, retail bank accounts, investment accounts, and other products. Not every API request needs to go through Compositor Services. For simpler APIs, requests are directly handled by Processor Services.

- **Processor & Transformer Services**
  A layer of microservices that performs most of the heavy-lifting work of our APIs. Here, request data from our APIs are mapped to non-standard messages and protocols that are spoken across various backend services, and responses are mapped back to standard API response format.
- **Connector Services**
  A layer of microservices that communicate with a variety of backend services typically using proprietary, non-standard messaging interfaces and legacy protocols and security standards.
- **Shared Services**
  A collection of components that are shared and used across many different microservices. These include databases, caches, message queues, service registries, logging servers, and encryption devices (e.g. HSM).
- **Cloud Infrastructure**
  All components comprising UniConnect Transact™ run within a Virtual Private Cloud that can be easily setup on most Cloud infrastructure. This provides security and isolation of all components running within UniConnect Transact™ application.
- **Backend Services**
  FI transactional systems, databases and applications often run behind several layers of firewalls. Through highly secure network communication methods, that combine VPN, IPsec, TLS, and end-to-end encryption, UniConnect Transact™ has the capability of talking to core bank systems and other data systems internal to FIs, from outside their private data-centers.

5. Our Approach to APIs

Percipient firmly believes that APIs are one of the most transformative technologies of modern times.

APIs as a computing technology is not new – in fact, it is as old as software itself. However, the latest iteration of APIs, which more specifically refer to Web APIs, has made it tremendously easy for software applications, written in almost any programming language and for any network connected device, to be able to talk to each other.

Percipient firmly believes that APIs are one of the most transformative technologies of modern times. This means it is an imperative capability for all FIs wishing to remain future-compatible. UniConnect Transact™ offers the ability to quickly build and securely expose at-scale APIs.

The popularity of APIs is also due to the fact that it completely lacks a set of strict rules and standards. Almost all previous iterations of APIs were burdened by complex set of myriad standards and protocols. As a result, implementations were often too cumbersome to use and adapt (e.g. EJB, WS-) or too difficult to interoperate across heterogeneous systems (e.g. RMI, DCOM), or too expensive (e.g. CORBA). With the benefit of hindsight, we feel quite comfortable in saying that the latest generation of

With UniConnect Transact™, developers have a choice of API formats that currently include REST-like and GraphQL, and in future, will additionally include gRPC.

API technology is set to become even much more pervasive in the coming years.

With UniConnect Transact™, developers have a choice of API formats that currently include REST-like and GraphQL, and in future, will additionally include gRPC. We believe that each of these three API formats have their own merits, and that offering just one of them is not sufficient for an upcoming generation of Fintech apps.

5.1. REST-like APIs

Most products in the market that offer APIs these days do so by implementing a design pattern known as RESTful APIs. Furthermore, the majority of APIs adhere to an API specification standard called OpenAPI Specification (OAS), which was previously (and are still more popularly) known as Swagger specification. However, more than 99% of such APIs are not in fact RESTful, because they don't support one of the key constraints of RESTful design, that is HATEOAS (OAS also does not currently support HATEOAS).

UniConnect Transact™ offers APIs that implement OAS 2.0 structures, along with a specification file (aka Swagger file) that can be easily imported into many programming languages and browsed online. Since our APIs do not support HATEOAS, we like to call them as REST-like because, along with almost all other APIs, they only partially support RESTful design guidelines.

5.2. GraphQL APIs

Besides a REST-like interface implementing OAS 2.0, UniConnect Transact™ also offers GraphQL. GraphQL or Graph Query Language, is an API scheme developed by Facebook, with the goal of giving control of API responses within client apps.

Whereas in a REST-like API, the data structure of API response is determined by the server (or service), in GraphQL, the client is in control and must query response data elements precisely. This makes GraphQL very powerful for certain type of applications, such as limited capacity devices (e.g. a smart watch app), or for applications such as chatbots.

5.3. gRPC APIs

gRPC is a more recent standard for APIs that was originally developed by Google. Its main goal is performance. Unlike REST-like and GraphQL APIs, both of which rely on text based messages, gRPC uses a binary messaging format that is very efficient over networks. gRPC also relies on a schema, which leads to lesser compatibility issues across client apps and server over longer periods of time.

6. Customer Journeys

UniConnect Transact™ APIs can be used to implement some very exciting customer journeys, across a range of channels and countless applications, including Mobile Apps, smart assistants such as Amazon Alexa, Microsoft Cortana and Google Assistant, chatbots, KYC, credit scoring, payments, and identify verification.

Here is an example of a customer journey that can be implemented using UniConnect Transact™ APIs.

6.1. Accounts Inquiry Customer Journey

1. Start with calling `POST /security/authentication/access-token`, submitting a customer's username and password. This operation returns an access token, which is used to get customer's data through other APIs.
2. After receiving an access token, call `GET/customermgmt/welcome/profile` and `GET /customermgmt/welcome/message` to fetch customer's basic profile and list of welcome messages.
3. Next, call `GET /accountmgmt/retail/accounts` to get a list of all accounts (with balances) that belong to the customer.
4. To get more detailed data for an account, including other balance types, you can call `GET /accountmgmt/retail/accounts/{id}`
5. To get a history of recent transactions for an account, call `GET /accountmgmt/retail/accounts/{id}/transactions`
6. Lastly, calling operation `GET /customermgmt/profile/profile` will return customer's full profile, which includes biographical, demographical, contacts, and other personal details.

7. Security

Percipient understands that the need for security cannot be compromised even amid the need to provide innovative customer services and solutions.

Ensuring security of data access, transmission and storage is of paramount importance across all banking and financial services companies. It is therefore natural to be concerned about security when it comes to exposing customer data through APIs.

Percipient understands that the need for security cannot be compromised even amid the need to provide innovative customer services and solutions. Therefore, not implementing APIs for security concerns is not an option any FI has today. Through a multi-layered security approach and thoughtful design, FIs can securely and confidently expose APIs, and reap the many benefits of Fintech engagement.

Security of APIs, data access and systems access is a huge topic and is beyond the scope of this document. However, below is a high-level summary of the ways in which UniConnect Transact™ handles security.

7.1. Access control via API Gateway

UniConnect Transact™ integrates easily with most popular API Gateway products, such as Azure API Management and IBM API Connect, among others.

In any API based platform, API Gateway plays a significant role in securing access. One of its key functions is to act as a gatekeeper and control who can access what APIs. UniConnect Transact™ integrates easily with most popular API Gateway products, such as Azure API Management and IBM API Connect, among others. These are some of the controls that UniConnect Transact™ relies on an API Gateway to provide:

- **Access policies** – defines who (a developer or an app) can access what APIs
- **Metering** – defines limits on how many API calls can be made by a certain developer or app
- **IP filtering** – defines which networks and subnets are allowed to access APIs

7.2. Network level security

As shown in the Overviewsection, UniConnect Transact™ has two primary network interfaces:

- **an inbound interface (using APIs)** – all communication through this network interface occurs over HTTPS and is secured using industry-grade TLS protocol and SSL certificates. We require a minimum support for RSA 2048 bits certificates and TLS v1.2. Optionally, two-way TLS (aka client-certificate authentication) can also be enabled for tighter security.
- **an outbound interface to communicate with backend services** – this interface is typically secured through various implementations depending on the type of backend service and FI environment. This would generally utilize a combination of VPN, IPsec and TLS. UniConnect Transact™ can additionally support message level encryption using various cryptographic standards such as PKI and symmetric key based ciphers.

All internal communications across various components within UniConnect Transact™ is secured through Cloud infrastructure level security controls and by creating a Virtual Private Cloud with IPsec security. This ensures that there is complete network-level isolation even when UniConnect Transact™ is deployed on a public Cloud infrastructure.

7.3. Securing User data through APIs

Data accessed by Apps through APIs are ultimately delivered to Users. However, not all Users should have the same level of access to data. It is also critically important

to ensure that User A cannot access User B's account data unless authorized to do so. APIs handle all this through a combination of authentication and authorization schemes.

UniConnect Transact™ supports OAuth 2 standard popular among REST-like APIs. It also supports role/attribute-based access control, using which each API can be defined to allow access only to certain roles or when certain User attributes match. Through these controls, companies can achieve very fine-grained control over User data access via APIs.

8. Software Development Approach

In this section, we address the methods and principles used to develop our products.

8.1. Agile and Scrum

Percipient develops product iteratively in Sprints that span over 2-weeks each.

Our product development teams use Agile development methodologies, together with Scrum model. We develop our product iteratively in Sprints that span over 2-weeks each. In each Sprint, we prioritize requirements, design, develop and test a small part of our software product, with a goal of delivering a fully tested code at the end of a Sprint. Multiple Scrum teams work concurrently in synchronized Sprints, with each Scrum consisting of roughly 6 full-stack engineers and 1 Scrum master. Features and requirements for each Sprint are managed through a 'backlog' concept, and we support changes in the backlog throughout a Sprint. We track our requirements backlog and task planning using Microsoft's Cloud based Agile project management tool called Visual Studio Team Services (VSTS).

8.2. Software version control

All good software development teams use a version control system for their source code and configurations. We use Git, a hugely popular version control tool in use these days. Our Git is hosted as part of Visual Studio Team Services (VSTS).

8.3. Polyglot model

Most software product teams choose one computer programming language (e.g. C/C++ or Java) at the beginning and stick with it throughout the life of the product. Once a codebase grows in size, it becomes very hard to move to another programming language. This leads to reduced adaptability and stifles continuous improvement of a product after the chosen programming language comes of age.

For UniConnect Transact™, we have adopted a polyglot programming model, where we develop parts of our product using different programming languages and techniques, and encourage use of new programming languages and framework as they become popular. Our APIs are currently developed using a combination of

Java/Springboot, JavaScript/Node.js and Python, and soon we plan to add Golang into the mix.

8.4. Continuous Integration / Continuous Delivery

For UniConnect Transact™, our teams practice continuous integration and continuous delivery, written in shorthand as CI/CD. In summary, we rely on coding and automation practices that build and compile our code, execute test cases, and deliver working code into our Test environments on a continuous basis, running several times throughout a day. These practices ensure that code-merge conflicts across multiple developers working on the same component are identified and resolved early on. It also minimizes bugs as software gets tested as they are being written. Finally, automating all of this reduces any effort wasted in performing repetitive tasks.

9. The future of the API economy

Percipient's UniConnect Transact™ and UniConnect Integrate™ platforms are designed to support banking and insurance-specific data systems, including the relevant metadata, logical and where available, canonical data models.

The API economy is well and truly in place and by all accounts, here to stay. This means that there will be ever-increasing demand for mashups across identity, transaction, location, social media, and events data into single end-points,withAI thrown in.

And while new APIs mean new opportunities, they also introduce new complexities. This applies as much to the data as the technology. As such, domain-specific data modeling resources now sit alongside API development capabilities as key requirements for app-building. Percipient's UniConnect Transact™ and UniConnect Integrate™ platforms are designed to support banking and insurance-specific data systems, including the relevant metadata, logical and where available, canonical data models. Our key objective is to ensure that FIs can launch customer applications with minimum delays, and FI and Fintech collaborations can hit the ground running.