

# **FINXACT CORE AS A SERVICE**

Innovation at the Core of Banking



**Finxact**<sup>™</sup>  
CORE AS A SERVICE

## FINXACT INTRODUCTION

Led by pioneers in the industry, Frank and Mike Sanchez, Finxact has built the first enterprise class cloud-native Core-as-a-Service platform that meets the performance, scalability, and regulatory requirements of the largest US financial institutions. Finxact's modern API first architecture enables banks to rapidly deliver new experiences by creating products on demand and integrating new services as needed. The Finxact team has been at the forefront of banking software for 35+ years, and together with its best-of-breed partners is disrupting core banking so financial institutions can truly transform on their terms.

## CORE PRINCIPLES



### Customer First

Finxact has one north star – its customer's success. This requires full dependability and transparency of its technology, commercial terms, and plans.



### Open Accessibility

Finxact's Core as a Service is 100% accessible via deep and broad APIs. Banks can rapidly integrate and create innovative customer experiences without technical or commercial barriers.



### Quality

Ensuring mission-critical stability while leveraging modern technology and delivering compelling economics – at any scale.



### Value

Focusing on engineering excellence means delivering advanced innovation and product ingenuity aimed at driving 10X improvements in its customers' most relevant metrics. Engineering excellence comes from deep core banking domain experience.

## The Right Domain Expertise

In the mid-90s, Frank and Mike Sanchez' company, Sanchez Computer Associates, introduced the first real-time core banking solution on the eve of Internet Banking. Early Internet banks included Bank One, ING, and American Express, among many others.

Their success led to an IPO in 1998, before being acquired by Fidelity Information Services Inc (FIS). After leading Enterprise Banking and Global R&D at FIS for many years, Frank co-founded ZenBanx, a digital-only bank which would go on to be acquired by SoFi. While developing the bank's platform he and his team of engineers concluded that even Profile, which was then the most advanced core banking system on the market, was not able to support the level of innovation possible with modern cloud platforms and container based applications. This led Frank, with encouragement and support from bank industry icons, to found Finxact with his brother Mike in late 2016.

Much of Finxact's team has two or three decades of domain experience building, implementing and operating core banking systems at scale. A demonstrable track record of success delivering complex, mission-critical, regulatory compliant core systems to US banks is a non-negotiable vendor criteria for banks looking to convert off their legacy systems. The Finxact team is one of the few companies in operation today that bring the credentials and credibility for that assignment.

“When we looked to build out our next generation digital bank, there were no suitable core systems.”

**Neil Underwood,**  
**President, Live Oak Bank**

## The Right Investors

Finxact's funding comes from strategic investors, several of which are its regional bank clients that have a long term interest in the success of its platform. Notable among its investors is the American Bankers Association, the 140 year-old institution that signaled to the industry by its investment in Finxact's that the company's approach to core banking was much needed in order to keep banks competitive. Finxact intends to remain committed to the first principles that compelled these iconic institutions to support its mission.

## The Right Approach

Finxact is disrupting the core banking market with its next-gen technology as well as its commercial terms.

The technology is not an upgraded core system. Finxact began development in early 2017 with no existing code, modules, or routines of any kind. The elite engineering team elected to build in the Go language created by Google for its high speed, performance and security. Kubernetes orchestrated containerized applications allow for the elasticity of the cloud to be harnessed and for the Finxact Core to be consumed as a Service.

Finxact's commercial terms present a stark contrast to the constraining contracts offered by legacy providers. For too long banks have been encumbered by contracts that impose exclusivity requirements, prohibitions against integrating with third-party solutions, and punitive termination clauses. This has greatly harmed the ability of banks to remain agile in their product offerings and competitive against non-bank fintechs. Finxact is an open core that is by design intended to allow for efficient API connectivity to the bank's third party vendors of choice and internal applications. The Finxact core is delivered As-a-Service, meaning bank clients pay only for what they consume without requirements of exclusivity.

---

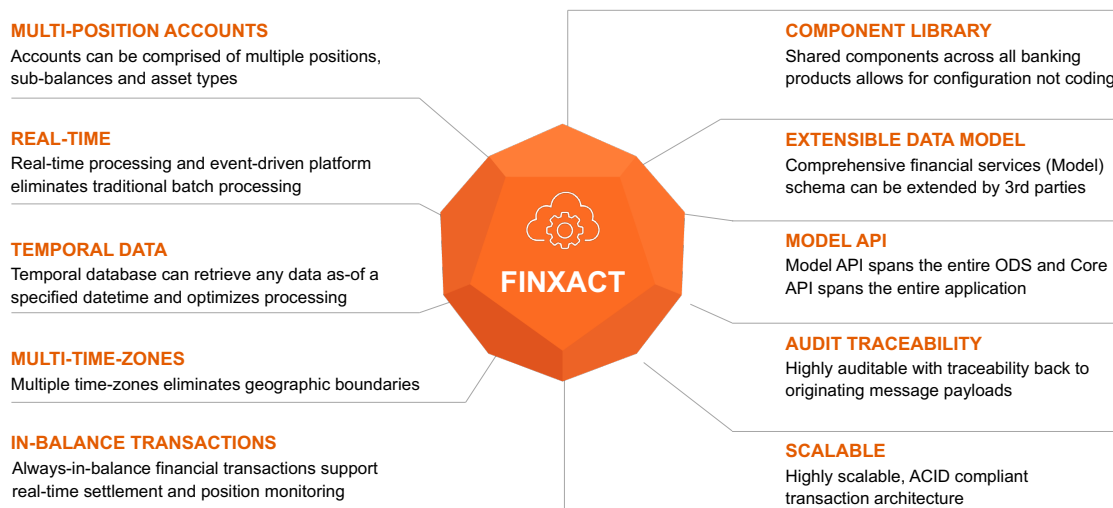
"We looked at most of the major core players, but Finxact was the only one that had exactly what we were looking for – a real-time, cloud-native headless core."

**Christian Winward,**  
**CIO, FirstBank**

## The Right Solution

Finxact is a highly scalable, real-time position keeping platform and transaction processing engine. It includes a comprehensive and extensible financial services model exposed as API endpoints. Its primary function is that of System of Record (SoR) for all of the transactions or events that affect the bank's deposit and loan positions.

The API-first design makes the SoR 100% accessible for banks to rapidly integrate solutions, whether they are sourced from third parties or developed in-house by the bank. Pre-integrated partners provide banks with even faster time to market. This approach allows the bank to retain maximum choice and control over its product ecosystem.



1

## Transformation Options

At most of the midsize and large financial institutions in the United States, the CIOs and CTOs are well aware of the pressing need to modernize their legacy core systems. As an industry, we have reached the point where the existential risk associated with not doing so now outweighs the attendant risk of project execution that typically accompanies core conversions. Generally speaking, the destination is an agreed-upon good place to get, but how to begin the journey and the path taken to get there is a question best answered by an individual bank's goals, needs, and timeline. The nature of the Finxact platform allows for multiple ways to begin the journey, with each step of the way supporting near-term business objectives that ultimately lead to transformation.

**Conversions:** Finxact's Core as a Service (CaaS) gives banks new options for how they succeed with a modern core conversion. Finxact's elastic scale and consumption-based pricing allows banks to begin conversion projects with minimal up-front capital requirements. Finxact's APIs allow banks to quickly integrate the necessary parts of their existing stack, running parallel, and rapidly configuring Finxact's core with the bank's banking products. Finxact's product-agnostic core means banks can start with a subset of products and over time, rapidly configure all of its deposit and loan products for migrating customers onto the new platform. The lower costs, API-first enabled integration, and rapid product configuration gives banks a new alternative to high-risk, high-cost conversions.

**Shadow Core:** Another transformation approach uses a concept that many large banks already have in place but with a twist. Banks often use an Operational Data Store (ODS) as a shadow database of one or multiple cores which authorize and accumulate transactions from multiple banking channels. For banks with batch-based (non-real-time) cores, this database does not negatively impact the timeliness of transaction updates since these systems update overnight and the ODS has accurate intraday running balances required for transaction authorizations. Using Finxact's native schema as the ODS gives banks a Shadow Core to support migration.

One of Finxact's customers reduced the number of API calls required for its New Account Opening process from 62 to 1 with Finxact.

The legacy core remains the bank's SOR while Finxact, acting as a Shadow Core, would process omni-channel banking activity and mirror the product behavior of the legacy core being replaced. The legacy and Shadow Core can use the same transaction information, running in parallel for testing purposes across cycles. Payment systems and data feeds can be tested without disrupting the production environment. Once calculations and product attributes are tested, the legacy core is "turned off" and Finxact's CaaS, using the native schema of the Shadow Core, becomes the new SOR.

**Parallel Bank (aka Digital-Only Initiative):** Finxact's CaaS is elastic in terms of scale, function, and commercial terms. This means banks may start their transformation with a narrowly-scoped, digital-only bank, adding products and customers over time. Banks can use Finxact's core for all of their deposit and loan products, growing the platform in product scope and customer scale. Banks with an end-goal to convert may also integrate some or all of their legacy stack, allowing banks to migrate legacy customers with less risk, cost and disruption.

---

Finxact is the first and only Core provider to become a member of the Open API Initiative whose other members include IBM, Amazon, Microsoft, and thousands of other companies.

## INTRODUCING THE FINXACT CORE

### Architecture

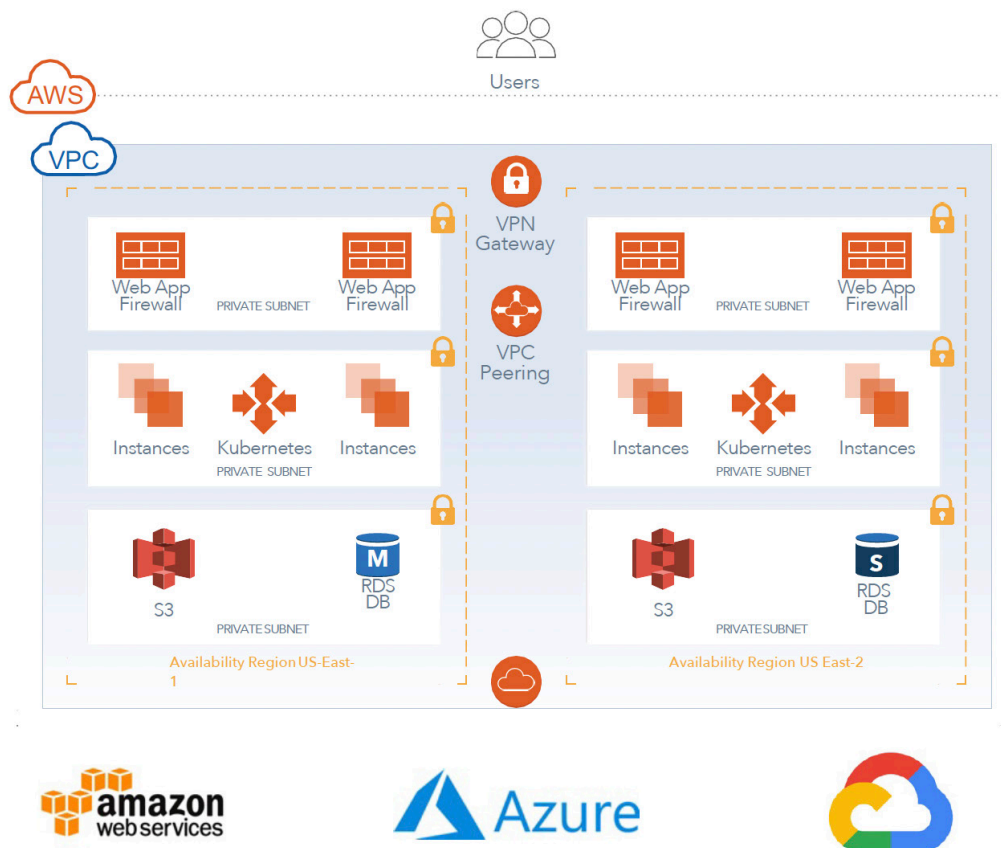
Finxact's Core as a Service is a real-time, straight-through application built on a cloud-native architecture. It leverages a containerized application component which allows for flexible cloud-provider agnostic deployments. The application is authored in a combination of JSON Schema for declarative Business Model definition along with highly efficient Golang core components.

Finxact is designed from the ground up as an API accessible System of Record, delivered in two flavors; Model and Core. Finxact provides RESTful HTTP web services (Model API's exposing 100% of the Finxact schema and Core API's for discrete business functions).

The Model API exposes 100% of the Finxact Model schema as CRUDL services against the persistence layer. The Core API exposes the application business services (e.g., open account, post-transaction) that provides discrete business functionality. These APIs are extensible by banks and third parties, and are reviewed for inclusion into core functionality. All Finxact APIs are exposed via the OpenAPI Specification (a.k.a., Swagger).

Finxact uses an agile Continuous Integration and Continuous Delivery (CI/CD) development practice and pipelines to ensure each and every code and schema change is tested and deployed into shared environments as rapidly as possible. The result is a compact, portable and efficient set of containers that can be deployed a wide range of infrastructure options.

Finxact's production deployments leverage Kubernetes container orchestration on Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform-based clusters. This configuration supports autoscaling at the application process level based on real-time workload requirements. Finxact has also published a co-authored white-paper with Microsoft Azure which covers its Azure Cloud deployment strategy in detail. This document is available on [Finxact.com](https://finxact.com).



Along with the Golang based applications, Finxact leverages a message queue to manage scheduled and real-time events. The current configuration uses a containerized instance, RabbitMQ. However, because Finxact uses the Advanced Message Queuing Protocol (AMQP), it could leverage another queuing technology or a managed queue service. The decision to use a containerized instance of RabbitMQ was driven by RabbitMQ's simplicity, maturity, and massive proven scalability.

The persistence layer of the Finxact Core is designed to provide flexibility in database options. The current version of the core uses PostgreSQL. PostgreSQL is widely used in public and private clouds and has been optimized in Database as a Service offerings such as Amazon Relational Database Service (RDS) and Amazon Aurora.

## Data Modeling

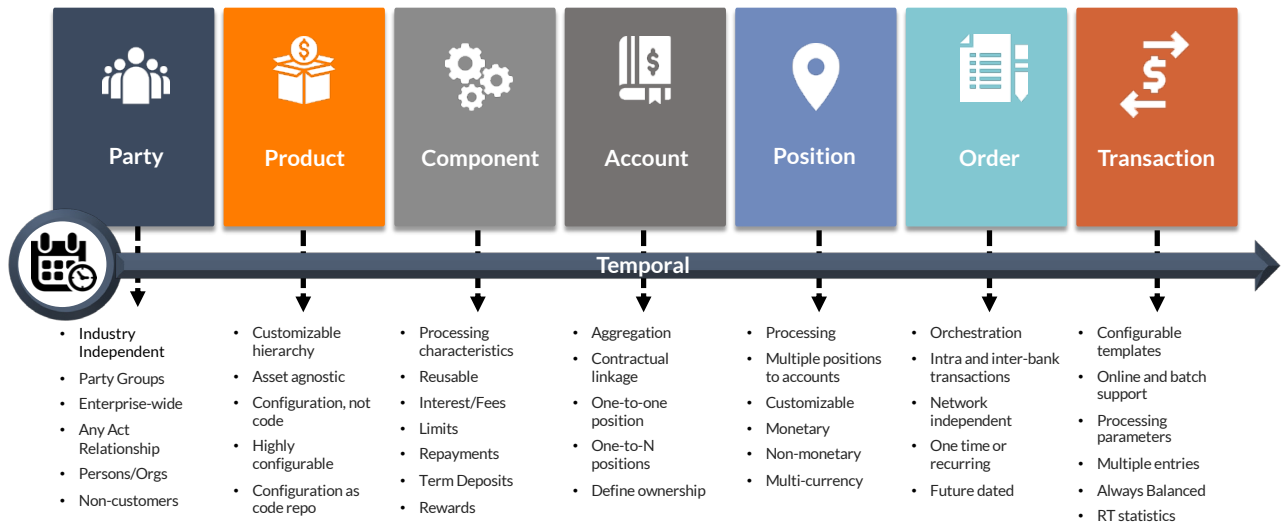
The Finxact SOR is a model-driven application, which is based on an object schema stored in a collection of JSON schema files. Finxact has extended the JSON schema to include metadata for database and application code generation requirements, including temporal storage features. The Finxact financial services model is object-based and hierarchical, and implements inheritance, composition, and templates. It defines abstract classes such as Party, Account, Position, Product, Transaction which are refined through descendants and composition components.



## Finxact Models and Entity Relationships

The Finxact schema files are processed by Finxact's proprietary code generator, which initially merges the base schema with optionally provided 3rd party schemas into a single consolidated schema instance, then generates application components, including the database DDL, the persistent classes, the model API endpoints and various documentation.

The Finxact persistence layer is 100% accessible through the model API (a collection of generated RESTful endpoints that implement CRUDL operations against the Finxact model objects, using an extended RFC6901 syntax). The model API provides a sophisticated and powerful object-based query syntax.



## Integration

Finxact provides RESTful API's exposing 100% of the Finxact schema and Core API's for discrete business functions. The Finxact Core also supports traditional file interfaces such as ACH and similar payment file formats to facilitate 'batch' style integration with legacy networks and systems. The file interfaces are available via the Finxact API to simplify access controls. In other words, there is no need to enable other protocols such as scp or sftp in order to upload and download payment files.

Integration with GL systems is supported via a file extract that can be configured to provide GL account level summary data or detailed GL transaction data. Another integration point in the Finxact Core is through the Event System.

The Finxact CaaS allows customer's systems, middleware and third-party solutions the ability to subscribe to application events (i.e. Financial Transaction or address change), facilitating real-time integration with a variety of applications such as fraud monitoring and customer messaging systems.



## Reliability

The Finxact CaaS can completely leverage the availability features of the underlying cloud infrastructure and DBMS. By deploying Finxact in a High Availability Kubernetes Cluster with failover that spans availability zones, Finxact supports 4 9's of uptime. Finxact leverages Database as a Service (DaaS) which provide High Availability across multiple Availability Zones. In support of disaster recovery scenarios, Finxact leverages real-time replication to a secondary availability region with a warm back-up. Finxact supports industry leading recovery-time and recovery-point objectives as part of its cloud-based deployments.



## Security

Finxact provides trusted and claim-based security including comprehensive entitlements (transaction authorization, centralized single mode RBAC and ABAC access control and fine grain permissions down to model property level). All protocols enforce API Security ([http](http://), ISO 20022, ISO8583).

All API messages request and response messages are saved in an application log (including identity and source information), which is referenced by the updatelog and all transaction history. Finxact includes a comprehensive schema package to manage user identity and login profiles, as an optional element in a configuration. API messages are required to contain the userRole userID, secret and potentially other requester context. Finxact supports federated security of API to enable pluggable authentication and identity management via its implementation with Okta.

As stated by Amazon AWS, running Amazon EBS encryption offers a simple encryption solution for EBS volumes without the need to build, maintain, and secure your own key management infrastructure. When you create an encrypted EBS volume and attach it to a supported instance type, the following types of data are encrypted:

- Data at rest inside the volume
- All data moving between the volume and the instance
- All snapshots created from the volume
- All volumes created from those snapshots

Encryption operations occur on the servers that host EC2 instances, ensuring the security of both data-at-rest and data-in-transit between an instance and its attached EBS storage.

As stated by Microsoft Azure, running their Azure Storage Service Encryption for data at rest helps you protect your data to meet your organizational security and compliance commitments. With this feature, the Azure storage platform automatically encrypts your data before persisting it to Azure Managed Disks, Azure Blob storage, Azure Files, or Azure Queue storage, and decrypts the data before retrieval. The handling of encryption, encryption at rest, decryption, and key management in Storage Service Encryption is transparent to users. All data written to the Azure storage platform is encrypted through 256-bit AES Encryption, one of the strongest block ciphers available.



## Scalability

The Finxact Core is designed ground-up as a highly scalable real-time transaction processing engine. The core posting logic has been optimized to absolutely minimize the amount of network and database activity required inside the 'critical path' transaction scope. The shorter the execution time inside TP, the less potential for contention for resources; especially important in a distributed update scenario. The Finxact technical stack, which includes the Alpine Linux OS and goLang programming language are also optimized for multi-threaded high-performance applications.

The Finxact Core can scale vertically and horizontally. In a vertical scaling scenario, a large single VM instance can host many worker containers alongside a database in the same memory space. This approach is desirable in scenarios where workloads are predictably high and relatively consistent. Horizontal scalability is more common and more easily achieved in the cloud. Specifically for the Finxact Core, worker containers can be hosted in a



---

## The Finxact core is designed ground-up as a highly scalable real-time transaction processing engine.

cluster spanning multiple VM instances. The cluster can be deployed with multiple tenants or exclusively to a single tenant. The core and reporting databases leverage Database-as-a-Service (DaaS) deployments which provide cost-efficient, resizable capacity. With AWS, Finxact is able to auto-scale based on increased utilization.

Finxact has developed code generation technologies to optimize the model to database interface (leveraging the features of the underlying DBMS) and to propagate any future optimization patterns rapidly across the application code-base. There are currently a number of heuristic database optimizations that the Finxact code generator recognizes when translating the Model schema into database DDL (e.g., handling of polymorphism, composition).

The Finxact core can currently generate an ANSI SQL compatible DDL implemented in SQL server and PostgreSQL.

Many model classes implement a database interface using a surrogate primary key defined in the schema as TGUID. This key includes datetime (microsecond precision) root followed by machine and process information that can guarantee local serialization as key and database level uniqueness, without any coordination requirements. The nature of the algorithm supports very fast key generation as well as optimal DBMS record insertion (minimized block management) and a PK index that optimized temporal database features (a Finxact application requirement).

Finxact has implemented temporal features in selected classes, which are defined via schema properties to facilitate 'as-of' (precise clock time) object selection. This allows the application to truly operate on a continuous basis without the need to generate snapshots or backups in support of time-sensitive processes. This feature, along with a 100% event-driven architecture ensures that all processing occurs in the correct relative sequential order and eliminates the requirement for end-of-cycle off-line or store and forward-based processing, (e.g., day/month/year-end). The elimination of cycle-based critical path batch or off-line processing in itself increases the operational capacity of the system.

Finxact has expended extensive thought and resources on its implementation of a temporal database (especially considering the throughput and scalability requirements of a real-time high TP-rate application), which is summarized in a white paper that can be provided if requested.

Finxact scheduling and event handling is datetime based, allowing events to be scheduled and/or cycled at a time level precision (i.e., in 302 seconds) as well as traditional date level scheduling (which relied on the external operational schedule of the process).

Finxact is designed such that the database interface is code generated based on a comprehensive and proprietary OO Data Model described in a JSON schema. The code generator produces the database interface via DDL and the class (struct) definitions in the goLang application language. This approach abstracts the core application from the underlying database technology (e.g., SQL, noSQL), and vendors and specific implementation features. As such, the application can leverage specific database scalability features if they exist, such as sharding, horizontal partitioning and hot-hot replication.

Finxact's CaaS is designed to be a highly scalable, high SLA transaction processing SOR, and is best suited as the API source for model acquisition and destination for persistence operations, with process extensive activities performed in an orchestration and/or workflow layer.



## Automation

The Finxact application is timeline and event-driven. All processes are scheduled on calendars which identify their execution time, therefore there is little operational involvement. For decision-based processes (e.g., fraud detection, AML), the system implements an embedded rules engine, that codifies a decision process. While Finxact does not provide cognitive behavior directly, such as AI, inference, or goal seeking, the rules engine can engage external applications via its embedded API capability (though must be respectful of response time/SLA, timeout, ACID compliance). Finxact rules can also publish events to either internal schedulers or external subscribers (e.g., enterprise case management and/or CRM systems).



## Transaction Processing

The Finxact persistence layer is ACID compliant. All messages are processed within a transaction fence that implements ACID compliance at the database level. Depending on the underlying DBMS, either granular level locking or optimistic concurrency control may be implemented, encapsulated within the TP strategy of the DBMS.

# FINXACT CORE AS A SERVICE – FUNCTIONALITY

## Customer Information Management (CIM)

The Finxact CIM model supports banking, e-commerce, CRM, and contact management applications. The core component is independent of any specific market segment, line of business or application requirements. It provides the following features:

- 3rd person view establishes Parties (e.g., Persons, Organizations) as separate entities independent of any relationship to any other entity.
- Design supports an incremental migration from existing systems and applications.
- Provide the ability to identify, measure, and manage customer exposure and risk across an enterprise, as well as across multiple enterprises.
- Increased profitability through informed business decisions spanning both internal and external relationships.
- Enterprise-wide platform for collecting all Party information from various systems, providing a unified view of each Party.
- Aggregated information from a single data source, enabling multidimensional analyses and increased capabilities for targeted marketing programs.
- Supports for multiple explicit relationship types, such as customer, household, affinity, family, etc., provides for much greater customer intelligence.
- Account aggregation recaptures customer and website loyalty away from third-party providers.
- Ability to facilitate transaction processing and workflow spanning across multiple systems (e.g., e-commerce, bill payment, funds transfers, cash management) improves efficiency and accuracy.
- Abstraction of schema from business processes provides flexibility to adapt processes to meet customer expectations.
- Ability to analyze other customer relationships such as billers, commercial, and retail affiliations. Centralized management of fraudulent Customer Id's, contact information and identification records substantially mitigates risk.



## Products

Any number of deposit and loan products may be defined, with an associated date offered and date expired. Products may use specific general ledger accounts or the general ledger accounts may differ based on account / position considerations, e.g., non-accrual, balance thresholds, etc. Products can be organized into a hierarchical organization that allows the bank to group specific types of products offered to specific types of clients. Virtually all product parameters may be modified on an account / position to tailor the account / position to meet specific clients' needs.

Because the Finxact Core allows for an account to be designated as a certain asset type it supports multi-currency account / positions and exchange transactions (single and cross currency) in any currency offered by the bank. Currency base rates and margins are defined in tables and maintained using Finxact Core's Bank Architect. An exchange transaction creates multiple accounting entries in order to support reconciliation and settlement, including offsetting debits and credits to currency positions and base equivalency positions for each foreign currency. These general ledger account / positions may be used by the bank to revalue or periodically "mark to market" foreign currency positions.

The Finxact CaaS is agnostic to asset type. Accounts / positions can be created with an asset type for example; loyalty points. Financial institutions can also use a multi-dimensional matrix to calculate and return values for any kind of Loyalty Points program.



## Deposits

Finxact's CaaS supports interest and non-interest-bearing accounts; fixed and variable interest rates; and multiple interest payment options, including capitalization, transfer to another account internal or external to the bank. Accounts may have terms and automatically renew (same or a different product type) or payout at maturity. Funds may be swept between accounts for any number of purposes, e.g., overdrafts, investment, to maintain specific balances. Cards and checks may be issued to provide customer account access. Checks issues may be uploaded to provide the bank with the ability to provide check production / distribution services for the customer. Holds, restrictions, stops, and reverse positive pay are all supported. Multiple positions within an account enable the Finxact Core to support laddered CDs, goal positions and savings bucket type accounts. Deposit limits may be placed per transaction, according to accumulations, regulatory requirements, bank risk, customer monitoring, and customer preference. Deposit overdraft sweeps can be enabled based on eligible products, target balances, minimum / incremental transfer amounts, one-to-one or one-to-many and can be bidirectional.



## Loans

Finxact's CaaS supports fixed P&I loans, both simple or calculated interest or variable payment loans, where the principal is a fixed amount or percentage of the balance and the payment includes accrued interest. Payments may be due at any frequency and rules based on the amount of the payment (full / partial), account status (due / past due), etc. determine how the payment will be allocated to principal, interest, and fees. Loans may be open or close-ended; and revolving or non-revolving. A loan repayment period may be different from the amortization period, e.g., a balloon loan. Interest rates may be fixed or variable, and minimum and maximum changes in the rate / payment may be set for multiple periods, e.g., per change or life of the loan. Loans may be secured or unsecured; cross-collateralized, or secured by one or more pledged collateral. Loan limits may be placed per transaction, according to accumulations, regulatory requirements, bank risk or customer monitoring.



## Relationship Pricing / Bundling Products

Finxact's CaaS defines an account as a contractual arrangement between the bank and its customer. An account may have one or more positions. Each position is a specific type of asset. A Finxact account may have multiple deposit positions, e.g., a USD DDA position and a CAD DDA position, a position for each of a lawyer's client relationships, a DDA position and a Savings position used to cover overdrafts, etc. An account may also have more than one loan position, e.g., a credit card account may have a position for purchases, a position for cash advance, and a promotional balance transfer position, etc. An account may also have a mix of deposit and loan positions, e.g., a DDA position and a LOC position, which may have its own check writing capabilities, but also cover overdrafts on the DDA.

Parameters used to calculate interest, assess fees, and set transaction limits roll up to components, and currently, components are linked to products, which drive the behavior of account positions. Finxact's CaaS allows components to be linked to entities other than products, which allows for much greater flexibility. For example, a limits component could be associated with a particular party allowed to transact on a position, e.g., CFO may have different ACH debit limits than a Controller.

Finxact's CaaS has a Business Rules Engine (BRE) that can be executed at numerous "interjection points," e.g., account / position onboarding and closure, pre and post-transaction posting, when calculating a fee, etc. This enables a bank to achieve customization it requires when not supported through the comprehensive parameterization options offered out-of-the-box. The Finxact BRE leverages a domain specific language (DSL) very similar to TypeScript.



## Interest Processing and Accruals

Finxact's interest accrual features are shared by deposit and loan products and include promotional rates, daily compounding, account based time zone processing, and closeout handling. Interest rates may be fixed, variable, tiered or multi-dimension (e.g., balance and term). Any number of interest indexes can be defined, with margins established based on the product type or customized for specific account / positions based on attributes of the customer / account / position. Minimums and maximums per rate change or for specific periods may be defined. Promotional rates are supported. The interest rate may be based upon account being funded in a bank defined time period. Balance upon which to accrue, i.e., ledger, collected, general ledger balance, etc. may be defined as well as accrual method, e.g., 30/360, actual / actual, actual / 365, etc. Optionally, interest may be accrued and charged for negative account balances.



## Transaction Processing

Finxact provides a design that allows banks the ability to define their own transaction codes. The transaction code controls the processing parameters, provides a template for multiple transactions entries within the transaction and acts as a vessel for ensuring balanced transactions. In other words, the transaction entries are the individual credits and debits within a single balanced transaction code. All transactions are always in balance and contain reference properties that link the individual entries together as well as the entire transaction to the original source record or message. The transaction allows for query logic with embedded formulas (e.g. a fee is x percentage of the transaction amount), as well as collecting statistics on usage for various reporting (e.g. limits, fraud, etc.).



## Payments

Finxact's CaaS provides order processing that supports intra and inter-bank money movement. These features are independent of any payment network, but the network used to process and clear the payment is part of the order itself. Orders may be one time or recurring, fixed or variable amounts, require maker / checker, and can be scheduled to retry when the account / position has insufficient funds or other conditions prevent it from being processed.



## Payment Gateway

The Finxact Core has been designed to support the real-time, or batch-based, processing across any applicable payment rail. The solution leverages its order model to orchestrate transaction processing across each configured network. Integration with respective payment networks is supported through its transaction processing APIs which support a series of plug-in interfaces including, but not limited to, ACH, X9, ISO20022, FAIM and ISO8583.

Finxact has also introduced the Finxact Payment Gateway which supports fully integrated connection, processing and settlement with the applicable clearing house. The Finxact Payment Gateway is provided as an out-of-the-box option to Finxact users in support of ACH, RTP and Wires. The Finxact payment API can also be leveraged, and extended, to support integration with any third-party payment network or rail (i.e., crypto-exchange, inter-bank networks, wealth-management and third-party payment hubs of any kind).



## Bill Payment

Bill payment services would be supported by third-party integrations in multiple ways. Finxact may be used to verify "good funds" before electronic transactions / checks are remitted by the bill pay application or electronic debits for the bank's customers can be posted at the time of remittance, while check payments may clear through normal processes. The approach utilized is dependent upon the capabilities of the third-party application and the bank's preferred approach to handling bill pay.



## Cards

Finxact provides basic card management functionality to support card issuance including card number generation and account linkage, card/account linkage management, card reissue, card end of life (expired cards), compromised card issuance with a new card number and expired card alerts. Finxact's CaaS supports embedded ISO8583 and an ISO20022 interface. ATM functions beyond these interfaces are provided by third parties.



## Check Processing

Check processing is handled via integration with companies such as Finxact's Ecosystem partner Deluxe (Wausau). Wausau produces outgoing X9 files, processes incoming X9 files and creates a posting file for the Finxact to process. Pay or Return decisioning will be provided by the Finxact's CaaS.

Finxact's accrual process is temporal by design, eliminating the need for interest buckets, batches and pseudo online processing.

The Finxact Core computes accruals on the fly on a position by position basis. The system first checks that all preconditions have been met on the position then computes accruals based on a predefined "as of" and "to" date and time down to the nanosecond. Consequently, there is no down time or other "smoke and mirrors" required to keep the system up and running. Just real time balances all the time!



## Fees & Commissions

Finxact's CaaS fee and service charge features are shared by deposit and loan products. Fees may be recurring, e.g., a monthly service charge, annual commitment fee, etc. or upon the occurrence of an event, e.g., a specific type of transaction, account opening, account closing, renewal, etc. Fees can be defined as a fixed amount or a percentage of an amount, e.g., transaction amount, account balance, etc. Tiered rate schedules are supported, e.g. first 5 are free and a \$.25 charge of any after the first 5. Fees can be collected from the account originated the fee or from another account. Limits may be established for a specific fee type for one or more time periods, e.g., daily limit of \$30, monthly limit of \$100. Fees in excess of a limit or where the account does not have a sufficient balance may be waived, take balance available / maximum possible, or the transaction may be rejected. Optionally, fees may be originated by third-party applications and posted through Finxact's transaction API. While Finxact does not currently perform escrow analysis natively. A credit report may be imported from a third-party application and used to offset fees assessed by the Finxact Core.



## Reporting

Finxact provides broad series of standard data extracts (ETLs) and reports that are operational, accounting, audit, and regulatory in nature. Optionally, extracts may be used as a source of data to third-party reporting solutions where custom reports may be created. Additionally, Finxact provides a real-time updated production replica operational data store (ODS) which can be leveraged by third-party reporting utilities, or any data extraction process, via Model API calls. The Finxact ODS enables full historical, point-in-time reporting via the solution's temporal data model. Finxact provides full access to data.

Finxact's CaaS may store / calculate data required for regulatory reporting, but not perform the actual reporting, i.e., information required to fulfill IRS reporting requirements is stored and accessible through extracts and APIs, but the Finxact Core does not provide paper / electronic forms for the customer or magnetic tapes as required by the IRS. The production of these forms / magnetic tape would be done by a third-party using data from the SOR.



## Reconciliation & Settlement

As a real-time SOR with a 'single version of truth' and 'always-in-balance', Finxact was expressly designed to require much less reconciliation than traditional batch systems. All transactions are always in balance and contain reference properties that link the individual entries together as well as the entire transaction to the original source record or message. All data from an originating EFT, ACH, Wire, Order, Bill-pay or any other source is captured (including terminal, journal Id, journal date, original message Id, settlement date, etc.), to facilitate balancing, match-kill reconciliation of clearing accounts and settlement processing.







## General Ledger

Finxact maintains persistent hierarchical general ledger accounts that reflect transaction activity in near real time. Stratification of a general ledger account is bank defined, e.g., balances are further segmented by cost center. Finxact's Core can create an extract to update a bank's general ledger on demand, i.e., there does not need to be a single extract at the end of the day.



## Regulatory & Compliance

Finxact's leadership team has long-standing involvement in the core banking business, and recognizes the importance of providing a solution that addresses security and risk controls as required by examiners, and other organizations like FFIEC, CFPB, OCC. Finxact continually reviews banking regulations to identify feature sets that will be beneficial and provide efficiencies in bank operations, like Reg E for funds transfers, Reg CC for availability of funds, and mechanisms to retain customer information in accordance with US Patriot Act/BSA and E-Sign Consent. Finxact carries out annual SOC 1, Type 2 and SOC 2, Type 2 third-party audits as part of its regulatory, security, and compliance posture. These documents are available upon request.



## Fraud

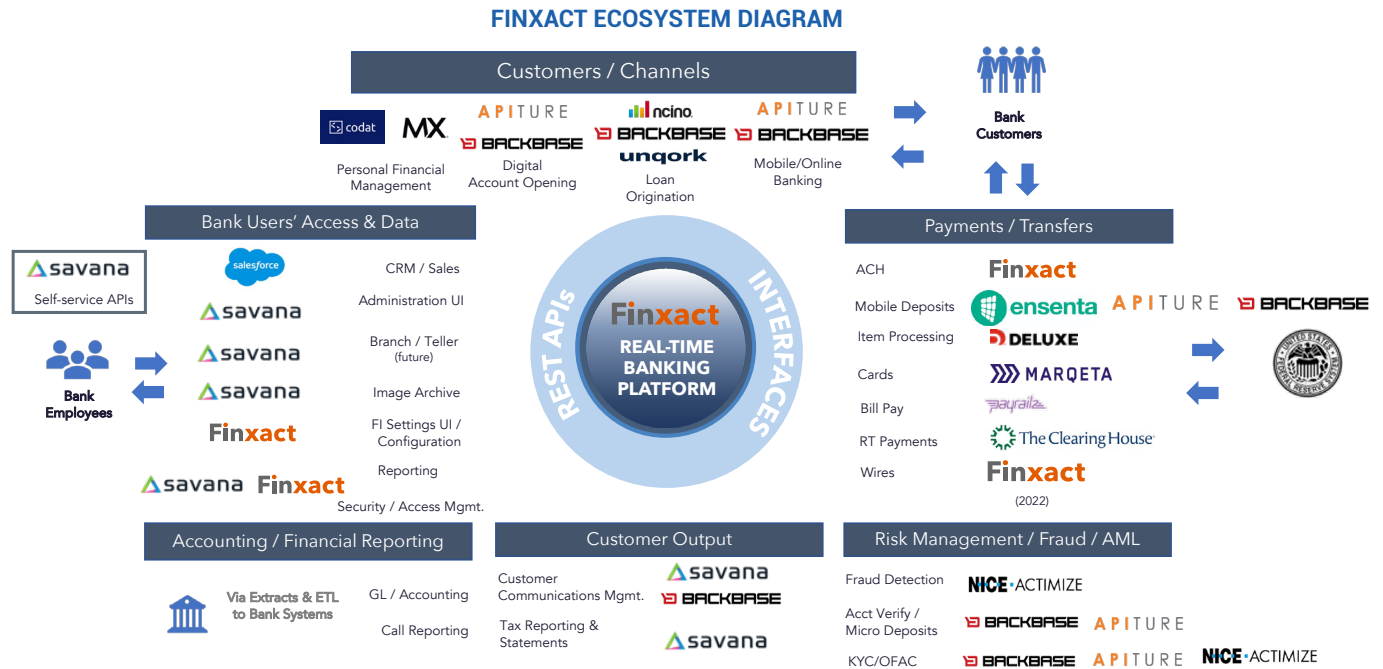
Finxact provides an embedded transaction processing engine that enables integration and coordination, with third-party fraud and AML prevention tooling of any kind. Transaction authorization and decisioning business rules may be authored to inject custom business logic throughout the transaction processing life-cycle. Alerts management supports multiple alert triggers e.g. scheduled events, file maintenance, limit violations, transactions, etc. Event queue integration with RabbitMQ enables external endpoints to subscribe to conditions in Finxact's processes and/or database for Fraud and AML detection.

---

Finxact maintains persistent hierarchical general ledger accounts that reflect transaction activity in near real time

# FINXACT ECOSYSTEM

Finxact provides banks with freedom to construct their banking stack, with a choice of pre-integrated Ecosystem partners, their own internal solutions, or other third-party providers. Finxact's early partners have demonstrated integrations in a fraction of the time needed to integrate into other cores.





---

WWW.FINXACT.COM | 866-834-6922 | 1301 RIVERPLACE BLVD. JACKSONVILLE, FL 32207

