



Bank P2P with Zelle

Creating an Integrated Payment Experience

By Scott Harkey • Payments Practice Lead

By Chris Hart • CTO

An Obvious Opportunity

Customers overwhelmingly prefer bank-provided P2P services. Given this, and the obvious customer demand for P2P services, P2P payments represent a strategic opportunity for banks to better serve their customers.

Emerging Threats

The rapid growth of non-bank P2P providers such as Venmo and Square threaten that opportunity, however. Regardless, large and mid size FIs should consider this opportunity as the new tables stakes, because conceding this experience will only dilute their value proposition in the eyes of their customers



Bank P2P with Zelle

Creating an Integrated Payment Experience

Executive Summary

Banks of all sizes continue to face threats from third parties interested in taking over key parts of the traditional bank customer experience. Many banks believe that conceding the customer experience to parties like Venmo, Google, and Square Cash will ultimately amplify the effects of disintermediation, further diluting the value proposition of banks in the eyes of customers and reinforcing their use of non-traditional services.

As a result, products like person-to-person payments are now table stakes for banks who want to ensure that customer interactions with their money occur through a channel and experience that they own. Given the nature of P2P, banks must work together to create ubiquity, so that any customer of any bank can send money to any customer of any other bank, and so that the experience is on par, if not better, than what these competitors can provide.

Zelle is the platform for creating that ubiquity; however, integrating it into a bank is no trivial task. Banks considering a Zelle integration should use this document as a high-level guide for identifying critical areas of focus and potential challenges upfront.

A Direct Challenge: Non-Traditional Payments

In today's rapidly evolving payments landscape, traditional financial service providers are on the defensive, experiencing regular attacks on multiple fronts. A key area where banks are being challenged is the P2P money transfer space. The number of P2P providers has increased dramatically, and more and more traditional bank customers are looking to third-party solutions to meet their needs.

Venmo, a popular P2P service owned by PayPal, experienced 174% year-over-year growth in 2015. Square Cash processed \$1 billion in transactions—within two years of launching. Given the obvious customer demand for P2P services, and the well-established trust that banks already enjoy from their consumers, P2P payments represent a key, strategic opportunity for banks to better service

A Unified Response

Zelle offers a P2P solution that is tightly integrated with existing banks' online and mobile channels. It also offers the chance for banks to work together to address two common challenges: a lack of ubiquity and a non-unified experience.

their customers.

P2P is not a new concept for most banks. Every month, millions of dollars are sent via existing P2P tools offered by major financial institutions (FIs). Many customers at large banks with transfer capabilities are familiar with the process of logging into their online banking (or mobile) site and sending money to a friend who is at the same bank. But what about when they bank at different places, or when the customer doesn't want to ask their friend where they bank? The big banks have solutions for this as well, but most customers either don't know they exist or don't know how to use them. Furthermore, they are not universally accepted.

For many users, the allure of using third-party P2P solutions is that they know it will always work, regardless of where the sender and the receiver bank. These bank-agnostic services employ different methods to make that happen: Venmo uses a stored value account, while Square Cash makes the receiver link a debit account to receive funds.

Whatever the method, they all provide the same value: facilitating the movement of money from person to person through a simple, elegant user experience. The challenge for banks, then, is to replicate that seamless experience and broadly communicate the interoperability of their service to customers across multiple banks. The answer to this challenge is Zelle.

A Joint Solution: Zelle

In 2011, Bank of America, Wells Fargo, and Chase joined together to form an entity known as clearXchange (CXC, now Zelle), with the goal of improving upon existing bank capabilities for transferring money between individuals. While the network ultimately leveraged traditional ACH files to trigger the movement of money, it added a shared directory structure and set of flows, operating rules, and agreements that allowed a customer at Bank A to transfer funds to a customer at Bank B in "real time," provided that both the sending and receiving banks were within the Zelle network.

Since that time, Zelle has added several other banks to its ownership, including US Bank, PNC, and Capital One. Its



acquisition by Early Warning Systems (EWS), introduced several other changes, including the addition of fraud detection and authentication services. Perhaps more importantly, it also brought a renewed focus on expanding the Zelle product beyond the initial core set of banks and on establishing a stronger brand awareness with customers.

The EWS merger brought a renewed focus on expanding the Zelle product beyond the initial core set of banks and on establishing a stronger brand awareness with customers.

Straightforward, Secure, And Fast: How Zelle Works

Before we dig into some common challenges banks face when implementing a Zelle-based solution, let's first look at how the solution works.

The core of the Zelle solution consists of a Zelle-hosted directory of users stored in the form of an email address or phone number (known as a token) and the underlying FI to which the user is linked. The Zelle directory facilitates the routing of transactions to the appropriate institution with only the user token, keeping all sensitive account and customer information within the FI that owns the customer relationship. Zelle also hosts consumer-facing online tools for users that are not part of a Zelle member bank; however, these are mostly used by customers who are registering to receive funds at a non-member bank.

In the most simplistic use case, a customer logs into his or her bank's online or mobile site and utilizes a "Send Money" function to trigger a Zelle-enabled transaction. How this is implemented varies slightly from bank to bank, but it usually consists of the following steps: registering the sender with the Zelle directory (which may happen behind the scenes), capturing the recipient's phone number or email address, registering that information with Zelle (if the recipient isn't already registered), then triggering a memo post to the sender account in order to deduct the amount sent.

The bank then preps a file that includes a Zelle-specific Payment ID in the addenda record, sending it to their ACH processing system during regular processing and using it to notify Zelle of the transaction. Zelle then notifies the receiving bank of the pending transaction, which notifies the receiving user of the pending payment and the timing of its availability. The receiving FI typically

Common Model, Unique Implementation

The Zelle customer experience requires integration with core banking systems. This integration can create obstacles for FIs that typically avoid IT customization or that rely on vendors to supply white labeled solutions, because every institution has a unique IT environment

uses an account number for a settlement account marked for all Zelle transactions. That FI is responsible for using the payment ID supplied in the addenda record to identify Zelle transactions and to transfer the funds to the customer's account for posting.

Addressing The Nuances

This "simple" use case does not address important nuances, however. While the fundamental Zelle model is fairly straightforward, use cases can quickly become complex in exception scenarios. Examples of such scenarios include when a receiver does not bank at a Zelle member bank or is already registered through a different bank than where they want to receive the money.

Zelle does provide a framework for addressing specific use cases in the form of Operating Rules, to which all member banks must comply. These rules detail all the basic use cases a participating bank must support, including: sending and receiving funds, account and token maintenance, and customer notification requirements. However, the FI must still determine whether they have their own company-specific scenarios or other considerations to address. This is in addition to the integration work that will be required.

Integrating With Zelle

It's common for banks to have questions about the integration required to support Zelle. When assessing the Zelle model, many FIs focus solely on the customer-facing components required. The impacts to a bank's mobile and online channels are straightforward. Likewise, the messaging requirement is easy to understand and is well within the capabilities of most banks.

Core System Impacts

The impact to the bank's core bank systems is less obvious, however, so it is important to understand why back-end integration is required at all. To illustrate this point, below are several common scenarios triggered by events within core bank systems that would affect the Zelle network:

- Processing ACH files to transfer funds received by a Zelle sender to the recipient within the institution



- Customer service requests to move a token from an old institution to a new institution
- Account re-numbering (e.g., due to fraud) triggering an update of a token to account mapping
- Account closure triggering de-registration of a customer’s token

Furthermore, integration with Zelle often occurs within the much broader context of bank systems. For example, most banks have some or all the systems represented in the diagram below (see Figure 1), many of which are impacted by one or more Zelle use cases.

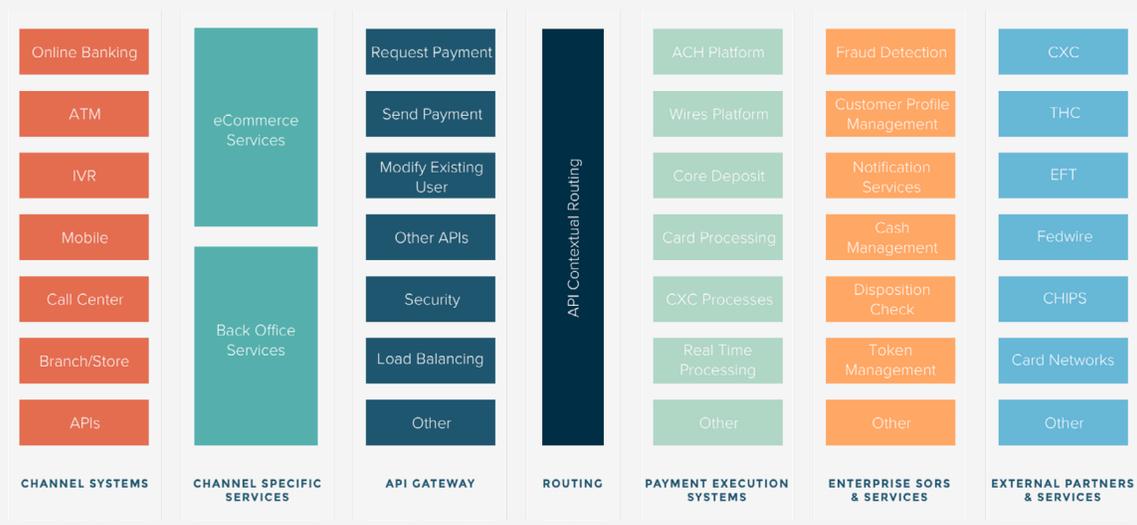


FIGURE 1.

Most banks have some or all the systems represented here, many of which are impacted by one or more Zelle use cases.

An implication frequently overlooked by many is the need to support both batch and real-time communication with bank systems. For example, when a customer at Bank A sends money to a recipient at Bank B, a real-time notification is sent to Bank B. The receiving institution is responsible for handling that message by generating an email or SMS alert to the recipient—in real-time.

Another requirement that some issuers find challenging involves the batch processing required by ACH files. When an ACH file is received that contains Zelle transactions, those records must be batch processed with additional logic outside of the ACH platform.

Data in the addenda is used to identify the Zelle transaction and transform the transaction to credit the recipient's account within the institution. Facilitating this process requires enhancing the bank's ACH system, core deposit system, or possibly both.

These are just a few examples of the types of events that frequently require core banking system integrations or enhancements but are often overlooked during an initial assessment of Zelle. In addition to these enhancements, new development is also typically required.

When assessing the Zelle model, many financial institutions focus solely on the customer-facing components required. The impact to the bank's core systems is less obvious, however, so it is important to understand why back-end integration is required at all.

New Applications and Services Required

As previously mentioned, integration with Zelle must be viewed within the broader context of a bank's technical infrastructure. We have already addressed how the bank's channel apps and core systems will be impacted, but the FI must also create new applications or services as well. These services are best understood in the context of specific Operating Procedures or Use Cases defined by Zelle, including:

- FI must register new tokens with Zelle
- FI must maintain the mapping of their customer's tokens to account
- FI must support returning funds sent to an unknown recipient after 14 days if the recipient does not register
- FI must handle exception scenarios for returned funds and claims
- FI must notify the non-bank customer of incoming funds if they are a new recipient and not part of a member-bank

Some of these services require new workflows and orchestrations that are specific to Zelle. Many of these interactions would need to occur between existing core systems. In some cases, the interactions may require modification to an existing enterprise system, or the creation of an altogether new one. They also create patterns that banks typically have not accounted for in their system architecture.

More Than Digital

Although P2P offering is primarily a digital experience, it does require customer service capabilities through a variety of channels. Most banks find one of the biggest challenges encountered when implementing P2P capabilities is the customer service scenarios for exception cases

The last use case listed above, where a customer sends money to a person that is not a member at another Zelle bank, is a perfect example. In this case, the sending bank must contact the receiver and provide instructions for registering with Zelle. This tends to present a challenge, as banks often have their notification system tied to a user profile, which is in turn often tied to an online banking ID or customer account. Since the receiver is not a bank customer, there will be no customer ID or online banking ID to which the bank can attach the user contact method. In addition to the technical challenge presented, this scenario also invokes privacy considerations, introduces new customer service scenarios, and requires legal and compliance approvals.

All these examples can be solved with technology and business process solutions, but they must first be identified and analyzed during the bank's assessment of the Zelle effort.

Another decision banks must make as they analyze the effort to build these services is which of them to build as Zelle-specific services and which to build as shared enterprise services for use elsewhere. The rapid evolution in the payments space and the regular introduction of new partners and opportunities leads many banks to consider more holistic implementations of any new functionality. Banks should perform the due diligence up front to assess the pros and cons of a shared services model from both a business and technical perspective.

Taking Care Of The Customer

Yet another area that gets overlooked until much later in the implementation timeline is customer service. While customers primarily utilize the Zelle functionality through online and mobile banking channels, there are still customer service implications to consider. We've already mentioned the messaging component of customer service, but additional use cases that require functionality include:

- A customer moving his or her relationship from one Zelle member bank to another may call the new institution requesting to move their Zelle token registration



- A customer may ask a customer service representative about the status of a Zelle transaction
- A claim may be created for a Zelle transaction that requires investigation and resolution

Financial institutions that integrate with Zelle should approach the effort with an end to end view of what it takes to operationalize the product and not simply focus on the initial technical integration.

Making The Solution Work For You

While every FI is unique, any FI seeking to integrate with Zelle can do the following to get a clearer picture of the effort required—and perhaps reduce it.

1 Perform a comprehensive architectural assessment of your payments systems prior to developing the Zelle-specific architecture.

Issuers often find that many of the new components needed for Zelle could also be used for other products, services, or partners. Looking at the functionality holistically helps to maximize the benefit of the build effort, but it must be implemented in such a way that it can be leveraged for other things. Performing this assessment prior to beginning the Zelle-specific architecture work will reduce the chances of being forced to change direction mid-effort to accommodate such opportunities.

2 Consider building a Token Management System/Database.

At a minimum, the Zelle integration requires issuers to track the Zelle Token and its related User, Account, and Preferences information. However, the use cases for a token management capability stretch far beyond Zelle. Similar token constructs exist in network-based card tokenization, future Real Time Payments efforts, and a host of other upcoming opportunities. Having a centralized utility for managing these tokens and their relationships to customers, accounts, or cards will enable several customer servicing and payment capabilities.

3 **Thoroughly investigate the capabilities of vendors and partners with regards to the specific integration needed for Zelle.**

There is no shortage of vendors that market the ability to integrate with Zelle “out of the box.” As mentioned, the number of integration points goes well beyond client front-ends and a few, self-contained processes; there are core banking system implications, customer servicing implications, and many other touch points for which no vendor can provide a generic solution. Some can provide a portion of the capabilities you require, but carefully analyze what is available versus what will need to be custom built first. Getting a clear view of this ahead of time will avoid underestimating the work effort.

4 **Carefully review ALL the required Zelle Use Cases prior to scoping the effort.**

Many banks begin the Zelle integration focused on the “happy flow” use cases, which can lead to improperly scoping the time and effort required. Negative or fringe use cases can be quite complex and often require a combination of technology, people, and process support. If these aren’t assessed early on, a nine-month integration effort can very easily balloon into an 18-month effort.

These are several of the major lessons that we learned while helping banks integrate with Zelle. While there are many other items that must be addressed, taking these into consideration should help you form a better understanding of the overall effort.

About Level

We are an IT, strategy, and design consulting firm that combines the innovative DNA of a start up with the wisdom, scalability, and process rigor of a Fortune 100 company.



Level Payments Practice

Level helps banks simplify and accelerate the adoption process for Zelle. Our Payments Practice includes members from the original team that designed the Zelle directory and implemented the Zelle solution at one of the original member banks. We have assisted other institutions in building their business strategy around P2P payments, assessing the effort to onboard with Zelle, and designing and building for a Zelle integration.

Level has helped four of the top five and nine of the top fifteen financial institutions in the United States with strategy consulting, product design, technology architecture, and solution delivery. The team understands the challenges banks face: a highly regulated environment, margin compression due to legislative changes, increased threats of disintermediation by new non-bank competitors, and legacy technology environments all make launching new capabilities challenging.

We can assist with all phases of a Zelle integration including: Initial

- Strategy assessment of P2P services
- Planning and estimating Zelle integration efforts
- Technology architecture for integrating internal systems and the Zelle directory
- Software development of custom components to support Zelle P2P capabilities
- Change management to prepare your organization's people, process, and technology for operating P2P services

Additionally, Level's technology partnerships with companies like Red Hat, Pivotal, and Docker means the solution Level delivers will work within your existing IT standards.