Appendix C – Scope of work

Overview

This Scope of Work is not intended to be all inclusive of the work TTX expects to be performed and is not a complete listing of all data elements or processes that may be required or desired. Essential requirements of the new application are:

1. Replicate the current business tax system data hierarchy
2. Same data fields as currently stored and UX and entry in similar organization
3. Naming convention and data format remain the same as current system

TTX seeks a functional twin of our current use of data application.

The overarching goal of this project is to transform our existing business tax system into a more agile, efficient, and custom-built solution that mirrors the unique operational workflows we have meticulously developed over time. We envision a replacement system that captures the essence of our efficient practices on a modern, custom-built framework that optimizes our daily operations.

This Scope of Work addresses the various modules and key information needed to understand the overall needs. Detailed and complete requirements will be provided by TTX and refined by TTX and the selected proposer during the project Discovery phase.

Business & Functional Needs

Due to other existing interdependencies, in addition to our core functional needs, there are key elements of the current state that cannot change and must remain in place with a new system.

Key Business Needs - elements of current state that cannot change.

1. **Core data elements:** Core data elements and associated naming conventions, formatting, structure and hierarchy.
2. **Connection & Integration with Online Tax Forms:** Connection to 20+ online tax applications, including associated API calls to and from the system.
3. **Data Integration & Sharing:** Existing views and extracts
4. **Configurability**

Key Functional Needs

5. **Web User Interface**
6. **Tax, Fee and License Calculations**: Existing calculations (tax amounts due, penalties, interest, etc.) and payment application rules.
7. **Payment Processing, Maintenance & Reporting:** Payment file formats, and associated processing functionality
8. **System Notes & Audit Trails**

1. Core Data Elements

All entities that meet nexus as detailed in the Business Tax Revenue Code are required to register with the Office of the Treasurer & Tax Collector. Each registered entity is assigned a unique seven-digit Business Account Number (BAN), and each registered business location for registered entities is assigned a sequential three-digit number, beginning with 000, and a unique 12-digit Location Identification Number (LIN).

The BAN houses all of the account's demographic information. Location specific information is stored on the various LINs that are on the BAN.

The "000" location on each BAN reflects its ownership information, primarily mailing address and ownership level taxes, fees, and licenses, such as the Gross Receipts Tax or Business Registration.

Each BAN also should have at least one "non-000" location. The 000 location stores further demographics related to the business structure. Sequentially, each physical location is assigned the next number (example: 001, 002, 003, etc.). These numbers are manually assigned by our users or systems and should be assigned sequentially per BAN. These locations in the system reflect the actual business locations. These locations may have taxes and/or licenses and/or fees associated to them.

Data elements for existing accounts, locations, taxes, licenses, and payments must be maintained and be unchanged.

Business Account Number (BAN) level information

- Business Name
- Organization Type
- Ownership Name
- All Trade Names/DBAs
- Full Tax ID for the business
- Mailing address (we currently only store one mailing address per account)
- Location information (each location has its own LIN, where more specific location information is stored)
- Business start date
- Business status (active/inactive)
- Business end date (if applicable)
- Business closure reason (if applicable)
- Combined Group Filer/Member information (including start/end date), if applicable

Location level information

- Situs address
- LIN
- Location Number
- Location start date
- Location end date (if applicable)
- Location closure reason (if applicable)

- Discovery date (date that the location was created in the system)
- Trade Name specific to the location
- On the 000 location only: phone number, email address

On each location, there could also be licenses and/or trust taxes.

Example

A successful bakery with two locations (one of which also has a successful parking lot) would look like this:

- **Business Account** (BAN) (houses business start date, business end date, business organization type, ownership names, trade names/DBAs, mailing address)
    - **Location 000** (LIN) (houses account contact phone, contact email address)
        - **2024 Registration Fee** (License Number, Bill Number, Tax Year, Registration Type, Basis for Fee, start date of fee to drive proration, calculated amount due, penalties, interest, fees, payments made, remaining amount due, details regarding the creation of the bill: who created it, how they created it and when, whether it was edited, etc.)
        - **2023 Registration Fee** (License Number, Bill Number, Tax Year, Registration Type, Basis for Fee, start date of fee to drive proration, calculated amount due, penalties, interest, fees, payments made, remaining amount due, details regarding the creation of the bill: who created it, how they created it and when/whether it was edited, etc.)
        - **Gross Receipts Tax 2022** (Trust Tax Account Number)
            - Period 1: (filing status, filing date, obligation details/integers, bill number, calculated principal amount, payments, penalties, interest, fees, remaining amount due, details regarding the creation of the bill: who created it, how they created it, and when/whether it was edited, etc.)
            - This tax type has 4 periods, each with the information listed above.
        - **Gross Receipts Tax 2023** (Trust Tax Account Number)
            - Period 1: (filing status, filing date, obligation details/integers, bill number, calculated principal amount, payments, penalties, interest, fees, remaining amount due, details regarding the creation of the bill: who created it, how they created it, and when/whether it was edited, etc.)
            - This tax type has 4 periods, each with the information listed above.
    - **Location 001** (Situs address, specific trade name, location start date, location system creation date)
        - Public Health license (License Number, Bill Number, Tax Year, License Type, Basis for Fee, start date of fee to drive proration, calculated amount due, penalties, interest, fees, payments made, remaining amount due, details regarding the creation of the bill: who created it, how they created it and when/whether it was edited, etc.)
        - **Parking Tax** (Trust Tax Account Number)

- Year 2023:
  - Period 1: (obligation details/integers, bill number, calculated principal amount, payments, penalties, interest, fees, remaining amount due, details regarding the creation of the bill: who created it, how they created it, and when/whether it was edited, etc.)
  - This tax type has 12 periods per year, each with the information listed above.
- Year 2024:
  - Same as above.

- **Location 002** (Situs address, specific trade name, location start date, location system creation date)
  - Public Health license (License Number, Bill Number, Tax Year, License Type, Basis for Fee, start date of fee to drive proration, calculated amount due, penalties, interest, fees, payments made, remaining amount due, details regarding the creation of the bill: who created it, how they created it and when/whether it was edited, etc.)
  - Fire Department license (License Number, Bill Number, Tax Year, License Type, Basis for Fee, start date of fee to drive proration, calculated amount due, penalties, interest, fees, payments made, remaining amount due, details regarding the creation of the bill: who created it, how they created it and when/whether it was edited, etc.)

Data Migration

A critical component of the project is the migration of data from the existing system to the replacement system. The vendor will be responsible for developing and executing a robust data migration strategy that ensures all historical data, including core data elements, calculations, and financial records, are accurately transferred to the new system.

Detailed discussions regarding data migration will occur during Discovery but expect the migration to include at least 100,000 active business accounts (each with 2+ business locations, and most with multiple tax obligations and payments), and at least 50,000 inactive business accounts (each with 2+ business locations, and many with multiple tax obligations and payments).

We will require at least 1 fully successful data migration prior to the "production" data migration.

Type	**Count**
Businesses & Locations	
Active Businesses	97,000
Inactive Businesses	147,795
Active Locations	122,000
Inactive Locations	184,222
Tax & License Information	
Active Tax Types Accounts – current year	26,000
Active outstanding obligations	150,000+
Tax Types	30+

License & Fee Types	400+
Penalty & Interest Calendars	45+

2. Connection & Integration with Online Tax Forms

Application Programming Interface (API)

Currently, tax filings are completed through online applications which make API calls to the tax system. This model will be continued in order to maintain the versatility afforded by segregation of the two components. Therefore, the current API calls between the tax application and tax system will need to maintain the existing layout. This approach will minimize downtime, streamline the migration process, and promote continuity in data flow and communication between systems. Vendors are encouraged to demonstrate their expertise in API integration and transition to meet this critical requirement effectively.

3. Data Integration & Sharing

The SF Business Tax data is critical economic data for the City. Significant portions of the data are used and shared internally to SF government and made publicly available.

Currently, there are more than 20 views and more than 30 regularly scheduled extracts that must be exactly replicated (format, layout, and frequency). Since most of these views and extracts support other software applications, the format must be retained. Specific details for the views and extracts will be shared during Discovery.

4. Configurability

In our current business tax system, our IT team has extensive configuration access and ability, including the ability to: create new taxes, licenses, and fees; update existing taxes, licenses, and fees; create new penalty calendars and update existing ones; create new rates and update existing ones; make mass updates via API.

It is a requirement of the replacement system that our IT team retain this ability.

5. Web User Interface (UI)

The Web User Interface (UI) is a critical component of the replacement system, serving as the primary means of interaction for our internal users. The UI should be intuitive, responsive, and designed with user experience at the forefront. Logical data elements should be collected and presented together.

The Web UI must be compatible with multiple web browsers, including but not limited to Google Chrome, Microsoft Edge, Mozilla Firefox, and Safari. Compatibility should be rigorously tested to ensure optimal performance across these browsers,

6. Tax, Fee & License Calculations

In addition to demographic information about registered businesses, the business tax system is the system of record for tax and fee filing information, including amounts due and payments.

Posting of Tax Filing data

The replacement system must be able to accept filing information from a variety of sources: from our existing online forms via existing API calls, as well as manually entered by our staff via the UI of a replacement system.

Regardless of the source or method of "posting" of the data to the system, the posting must calculate an "obligation" and assign a unique Bill Number to that obligation, following the convention currently used within our ecosystem.

If obligations are migrated to the replacement system, existing bill numbers for existing obligations must be maintained and cannot be changed.

Calculation of Tax/Fee/License amounts

Obligations are calculated in a variety of ways, all of which must be replicated within the replacement system. There are over 20 different tax types, each with its own rate configuration. There are over 250 different license types, each with its own rate configuration. Some examples are provided below to illustrate the various rates. Specific rates and requirements will be clarified and verified by TTX in Discovery:

Simple Complexity Tax Calculations

- [Taxable Amount] x [Specific Tax Rate] = [Total Tax Due]
    - Where Taxable Amount is input into the system either manually or via API, and then is multiplied by stored/configured Specific Tax Rate to calculate Total Tax Due.

Intermediate Complexity Tax Calculations

- [Gross Amount] – [Exclusions] = [Taxable Amount]
    - Where Gross Amount and Exclusions are input into the system either manually or via API, and the system performs simple math to calculate Taxable Amount, which is subsequently used to calculate Tax Due as illustrated below.
- [Taxable Amount] x [Specific Tax Rate] = [Total Tax Due]
    - Where calculated Taxable Amount is multiplied by stored Specific Tax Rate to calculate Total Tax Due

Advanced Complexity Tax Calculations

- [Gross Amount 1] – [Exclusions 2] = [Taxable Amount 1]
    - Where Gross Amount and Exclusions are input into the system either manually or via API, and the system performs simple math to calculate Taxable Amount, which is subsequently used to calculate Tax Due as illustrated below.
- [Taxable Amount 1] x [Specific Tax Rate 1] = [Tax Due 1]
    - Where calculated Taxable Amount 1 is multiplied by stored Specific Tax Rate 1 to calculate Tax Due 1
- [Gross Amount 2] – [Exclusions 2] = [Taxable Amount 2]
    - Where Gross Amount 2 and Exclusions 2 are input into the system either manually or via API, and the system performs simple math to calculate Taxable Amount 2, which is subsequently used to calculate Tax Due 2as illustrated below.
- [Taxable Amount 2] x [Specific Tax Rate 2] = [Tax Due 2]

  - Where calculated Taxable Amount 2 is multiplied by stored Specific Tax Rate 2 to calculate Tax Due 2
- [Tax Due 2] + [Tax Due 2] = [Total Tax Due]
  - NOTE: this illustrates 2 different rates within the same tax filing, but some of the tax configurations have 3 or more rates within the same tax filing.

Simple Complexity License Calculations

- EXAMPLE 1: [# Units] x [Specific Rate per unit] = [Total Amount Due]
  - Where # Units and license type code is input into the system either manually or via API, and then is multiplied by stored/configured Specific Rate to calculate Total Amount Due.
- EXAMPLE 2: [# Units] x [Specific Flat Rate] = [Total Amount Due]
  - Where # Units is input into the system either manually or via API, and then Flat Rate is triggered as Total Amount Due.

Intermediate Complexity License Calculations

- [# Units] x [Specific Tiered Rate] = [Total Amount Due]
  - Where # Units and license type code is input into the system either manually or via API, and then Specific Rate is applied to first designated count of units, followed by a different rate for the subsequent units.
  - Example: A license for Tow Trucks, where the first truck costs $100, but each additional truck costs $50. If the unit count entered is "4" then the system must be able to calculate: (1 x $100) + ((4-1)x$50)) = $250.
- [# Units] x [Specific Tiered Rate] = [Total Amount Due]
  - Where # Units and license type code are input into the system either manually or via API, and then the system applies a multi-tiered rate table to determine the Amount Due.
  - Example: A registration fee is based on Gross Receipts, so the Gross Receipts amount is entered as the "# Units". If Gross Receipts are 0 – 100,000, then the rate is $25.00. If Gross Receipts are 100,001 – 200,000, then the rate is $35. If Gross Receipts are 200,001 – 300,000, then the rate is $30. If the unit count entered is $102,000, then the system must be able to calculate $35 as Total Amount Due.

Calculation of Penalties, Interest & Fees

Detailed penalty, interest and fee calculations will be outlined and clarified during Discovery. High level overview as follows:

Penalty

- Penalty is assessed on unpaid obligations monthly, beginning the day after the deadline.
- In general, the current penalty structure is: 5% penalty (based on remaining principal amount in month 1), an additional 5% of penalty (based on remaining principal amount) in month 2, an additional 5% of penalty (based on remaining principal amount) in month 3, an additional 5% of penalty (based on remaining principal amount) in month 4, and an additional 5% of penalty (based on remaining principal amount) in month 5.

- Note that penalty is calculated based on the remaining principal amount at the time of the penalty calculation.
- In most cases, there is no additional penalty assessed in month 6 and beyond.
- Older obligations have a different penalty structure: 5%, 10%, 15%, 40%.
- We must retain the ability to adjust penalty amounts as needed.
- Additional penalty types also exist and must be configurable.

Interest

- Interest is assessed on unpaid obligations monthly, beginning the day after the deadline.
- In general, the current interest structure is: 1% interest (based on remaining principal amount in month 1), an additional 1% in month 2, etc.
- Interest continues to accrue until the balance is paid in full.
- We must retain the ability to adjust penalty amounts as needed.

Fees

- Late tax filings are charged a $55 administrative fee.
- Late registration payments are charged a $55 administrative fee.
- Additional fees, with different logic and calculation schedules, may apply.
- We must retain the ability to adjust penalty amounts as needed.

Surcharges

- Some licenses have specific surcharges associated with them. Those are configured to automatically charge any time that the specific license type is added.

Notes regarding all Rates & Calculations

Rates frequently change, so they must be configurable (by TTX IT staff) and effective dated. As an example, the rate for a license may be $25 per unit for Year 2024 and $28 per unit for Year 2025. The system must be able to calculate based on data element (example: unit count) and effective year/period so that the 2025 year license uses the 2025 rate and the 2024 year license continues to use the 2024 rate. This is also true for the tax types.

Historic rates currently configured in our existing system must be migrated and functional in the replacement system.

For taxes, the critical data elements are:

- Trust Tax account number
- Tax type
- Filing Frequency (which due/date penalty calendar applies)
- Start date for the tax (when the tax started applying to the location or business)
- Out of business date, if applicable (when the tax stopped applying to the location or business)
- Status (active/inactive)
- “Last period is total year recap” indicator (some tax types “true up” at the end of the year)
- Total amount due for all periods
- Total amount past due for all periods

- Different taxes have different numbers of periods (example: Quarterly tax types have 4 periods, monthly tax types have 12 periods, some annual tax types have 1 period). For each period:
    - Filing status (not applicable, not filed, filed, paid)
    - Year
    - Period
    - Filing date (drives some fees)
    - Bill number
    - Integers/elements of the tax (see "Tax Calculations" section of this Scope of Work)
    - Tax amount due *(calculated by system)*
    - Penalty amounts *(calculated by system)*
    - Interest amounts *(calculated by system)*
    - Fees *(calculated by system)*
    - Payment information
    - Snapshot as of filing date (principal, penalty, interest, fees)
    - Current balance (principal, penalty, interest, fees, payment information)
    - Date posted to system and by which user
    - Date corrected in system and by which user, if applicable
    - Previous filing information, if applicable

For licenses/registration fees, the critical data elements are:

- Tax Year
- License Number
- Bill Number
- Billing Date (date to start the penalty calendar calculations)
- Code (the short code associated with the specific license/fee type; each license/fee type has its own rate structure and calculation)
- Long description (each type has its own longer description)
- License status (active/inactive)
- Expiration date
- License start date (this date also drives proration calculations for principal amount)
- Unit Count/basis (called "Current amt/qty")
- Calculated "Tax" or principal amount
- Fees (late payment/administrative fee for late filings and/or standard fees per license type)
- Penalties (calculated per penalty calendar and calculator)
- Interest
- Applied payments
- Tracked surplus
- Total amount due
- Permit Number
- If closed, closure reason and closure date.

Adjustments & Corrections

The system should automatically calculate taxes due, penalties, interest, fees, etc. However, there are also scenarios where TTX needs to manually calculate and adjust penalties, interest, fees. In these cases, it is imperative that the user be able to select the SPECIFIC TYPE of penalty, interest or fee that is being adjusted (added, removed, or adjusted).

Payment Allocation & Application

Payments are allocated as dictated by the Business Tax Code, by obligation element type (principal, specific penalty types, interest, etc.). The allocation must exactly mirror the business tax code. The replacement system must auto-calculate ongoing penalty, interest and fees based on effective dated payments. This requires that the system calculate penalties and interest daily on delinquent obligations, but also be able to reverse out penalties/interest, and apply payments, based on a payment effective date provided in the payment file.

7. Payment Processing, Maintenance and Reporting

Payment Information Load

All payments are posted into the system via a file that is generated by the centralized remittance software application. Currently, the loading of the file is time-consuming, tedious, and error-susceptible six step manual upload process. In the replacement system, payment data will continue to be sourced from the remittance system. However, the replacement system should use a non-manual, less time-consuming and more error-proof process to ingest it.

Payment file must match a pre-defined file layout, which will be provided in Discovery.

NSF processing

The replacement system must post negative transactions and associated costs for insufficient funds as well as re-calculate penalties and interest due to the failure of payment.

Refunds

The replacement system must support our existing manual and automated (Robotic Process Automation) refund processing. (High level overview of current process: Search by PIN/Bill Number, verify surplus amount, code as refunded and assign a unique, system-generated “refund number.”)

Money movement

The system must support the transfer of funds from one obligation to another. When money is moved from one obligation to another, it must maintain its original effective date, trigger recalculation of penalties and interest on both the obligation it was moved *from* as well as the obligation it is moved *to*.

8. System Notes and Audit Trails

Notes should be able to be added and date/time stamped. All money movements and account changes should be noted automatically and dated.

Transaction Volume

Transaction Type	Primary Method	Average Annual Count
Account Creation & Updates		
New Business Registrations per year	API	12,000
New Location Registrations per year	API	5,000
Business Closures per year	API	7,000
Location Changes per year (Open/Close)	API	10,000
Contact Information Updates	API	5,700
Tax Filings		
Online Forms	API	125,000+
Direct Entry into System (UI)	UI	10,000+
License obligation postings	API	15,000
Manual Adjustments	UI	10,500
Payments		
Payments Processed	Upload File	178,580
Refunds processed	RPA	3,000
	UI	2,500

Architecture

Our current system is an on-premises solution. TTX is seeking an architecture that aligns with industry best practices and modern technological standards, ensuring the system's scalability, reliability, and maintainability. This architecture should support the existing components of our system, including databases, user interfaces, APIs, and scheduled jobs. To foster adaptability and accommodate evolving technology trends, the architecture should also be designed with futureproofing in mind.

Databases & Database Views

The database architecture of the replacement system is critical to the success of this project. It must facilitate efficient data storage, retrieval, and integration with other system components while maintaining compatibility with our existing data structure. The databases should be designed to retain the core data elements, hierarchy, and naming conventions as found in our current system, as outlined above in Key Business Needs.

The replacement system should utilize database technologies and configurations that optimize performance, scalability, and security while aligning with industry best practices. The architecture should also allow for the migration of existing data, ensuring the continuity of core data elements and calculations in the new system. TTX will work collaboratively with the selected proposer during the Discovery phase to refine database requirements and design for maximum efficiency and compatibility within our broader technology ecosystem.

Design a database structure and implement mechanism to store active data and moving historical data to a separate database and purge based on retention and regulatory requirements.

The city uses Snowflake as a standard data platform. It manages and shares large data volumes securely and supports collaboration across departments for data analysis.

Application Server

The application server plays a pivotal role in ensuring the performance, scalability, and security of the replacement system. It should be designed to handle concurrent user requests efficiently, with a focus on delivering a responsive user experience. The server architecture should also accommodate the expected system load and be designed to scale as our business needs grow.

Security measures, such as authentication, authorization, and data encryption, should be implemented at the application server level to safeguard sensitive information and protect against potential threats. The application server should be designed to support scheduled jobs, batch processing, and other critical system functions.

Scheduled Jobs

The replacement system must support scheduled jobs and batch processing to automate routine tasks, data updates, and system maintenance activities. Scheduled jobs are crucial for optimizing system performance, ensuring data accuracy, and automating repetitive processes. The replacement system should provide the capability to define, schedule, monitor, and manage these jobs effectively.

Scheduled jobs may include but are not limited to:

1. Data synchronization with external sources.
2. Periodic data backups and archiving.
3. Report generation and distribution.
4. Database maintenance tasks.
5. Compliance checks and audits.
6. System health monitoring and alerts.

It is imperative that the replacement system offers a user-friendly interface for configuring and scheduling these jobs. Additionally, the system should log job execution details, provide alerts in case of failures, and offer options for job prioritization and resource allocation.

Security (Disaster Recovery)

Ensuring the security and resilience of the replacement system is critically important to safeguard sensitive data, maintain business continuity, and mitigate potential risks.

Disaster Recovery

The replacement system must have a robust Disaster Recovery (DR) plan in place to ensure that essential business functions can be quickly restored in the event of system failures, natural disasters, or unforeseen incidents. The DR plan should include:

1. Regular data backups with versioning.
2. Off-site data storage to prevent data loss.
3. Redundant hardware and infrastructure to minimize downtime.

4. Clear procedures for system restoration.
5. Regular testing and updates of the DR plan.

The goal is to minimize downtime, data loss, and disruption to business operations in the face of adversity.

Hosting

This section outlines the requirements and expectations for the hosting of the replacement system. It is essential to have a robust hosting solution that ensures the system's availability, performance, and security.

Hosting is a critical component of our long-term vision to maintain and enhance our operational practices. We are not only looking for a hosting solution but also a strategic partner who understands our overarching objective. The chosen hosting provider should align with our goal of ensuring a robust and scalable infrastructure that supports our efficient workflows and operational agility.

Hosting Solution

The hosting solution must meet the following criteria and set up in one of the existing City cloud tenancies at Google Cloud Platform (GCP), Amazon Web Services (AWS), Azure and Oracle Cloud Infrastructure (OCI).

- **Reliability** The hosting environment must guarantee high availability and minimal downtime. It should provide redundancy and failover mechanisms to ensure uninterrupted service.
- **Scalability** The hosting infrastructure should be scalable to accommodate potential growth in data and user activity. It must handle increased load without compromising performance.
- **Security** Robust security measures must be in place to protect the system from cyber threats and data breaches. This includes firewalls, intrusion detection, encryption, and regular security audits.
- **Performance** The hosting environment should offer optimal performance to ensure fast response times and efficient data processing. It must be capable of handling peak loads without degradation.
- **Data Backup** Regular automated backups of all system data, configurations, and logs are required. Backups should be stored securely and be readily accessible for disaster recovery purposes.