



Integration Guide

Version History

Date Changed	Version	Author(s)	Changes
Nov 1, 2023	D2	Sean Dulac (Appko) Paul Esch (TSANet)	First draft version
March 27, 2024	D3	Sean Dulac (Appko) Paul Esch (TSANet)	Added Notes API and Swagger doc location
May 25, 2024	1.0	Sean Dulac (Appko) Paul Esch (TSANet)	API updated, Salesforce reference updated
June 13, 2024	1.1	Sean Dulac (Appko) Paul Esch (TSANet)	API completed to include notes and response update.

Table of Contents

Version History.....	2
Contributors and References.....	3
Executive Summary.....	4
Benefits of TSANet Connect Integration.....	5
TSANet Connect – Integration Framework	7
TSANet Connect – Database Structure.....	8
TSANet Connect 2.0 REST API.....	9
Endpoint (POST): login.....	9
Endpoint (GET): me	9
Endpoint (GET): partners (search).....	10
Endpoint (GET): form/department.....	10
Endpoint (GET): form/documentID/field	11
Endpoint (GET): form/documentID/field/selections	12
Endpoint (POST): cases/create	12
Endpoint (GET): cases/(InternalCaseNumber).....	13
Endpoint (GET): cases.....	13
Endpoint (POST): case/approve	14
Endpoint (POST): case/update/approval	14
Endpoint (POST): cases/notes/create	15
Endpoint (GET): cases/notes.....	15
Integration Process	16
Reference Integration – Salesforce.....	17
Salesforce Install Guide	17
Salesforce package developer guide.....	17
Salesforce Package Components.....	17
User Experience – Outbound Request (To Member).....	20
User Experience – Inbound Request (From Member).....	21
Ongoing Updates – Collaboration with Connect 2.0 Members.....	22
Ongoing Updates – Response and Notes Feature.....	23
TSANet Case Status and Priority.....	24

Contributors and References

Contributors

Paul Esch, TSANet, <https://www.linkedin.com/in/pauleesch/>

Sean Dulac, Appko, <https://www.linkedin.com/in/seandulac/>

References

The following documents explain the details behind the integration framework Technology and Methods.

Enterprise Integration Patterns: Designing, Building, and deploying messaging solutions.

By Grregor Hohpe and Bobby Woolf -

<https://www.amazon.com/o/asin/0321200683/ref=nosim/enterpriseint-20>

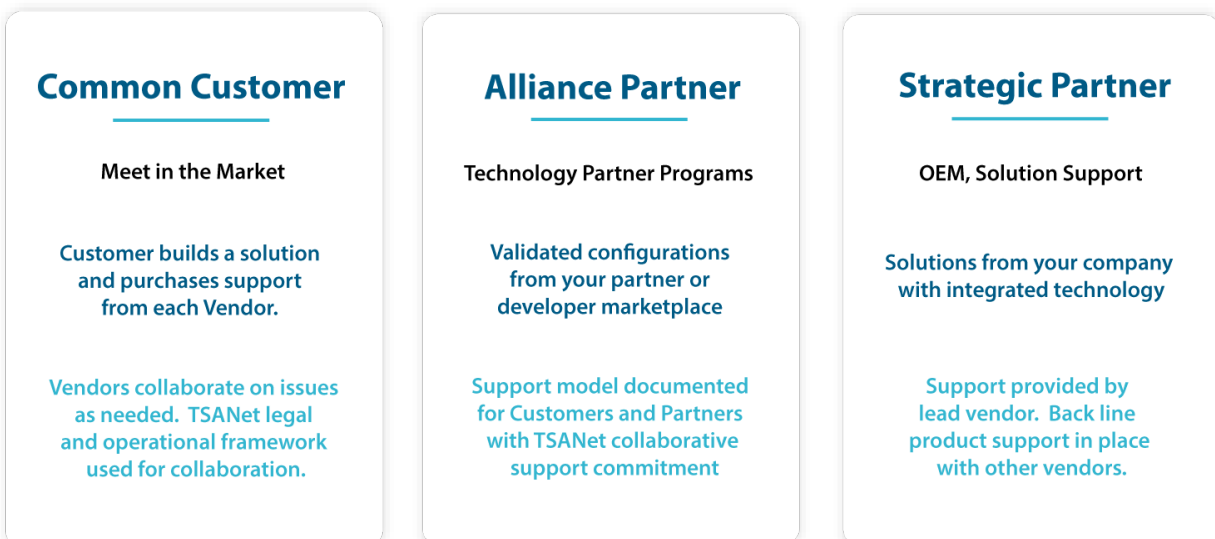
Apache Camel: <https://developers.redhat.com/products/redhat-build-of-apache-camel/overview>

Executive Summary

TSANet is a member-driven, not-for-profit, global collaborative alliance of 900+ technology companies working together to improve the support experiences of their shared customers. Member companies use TSANet to enable collaboration with their Technology Partners. The TSANet Partner Framework is flexible and supports several relationship models.



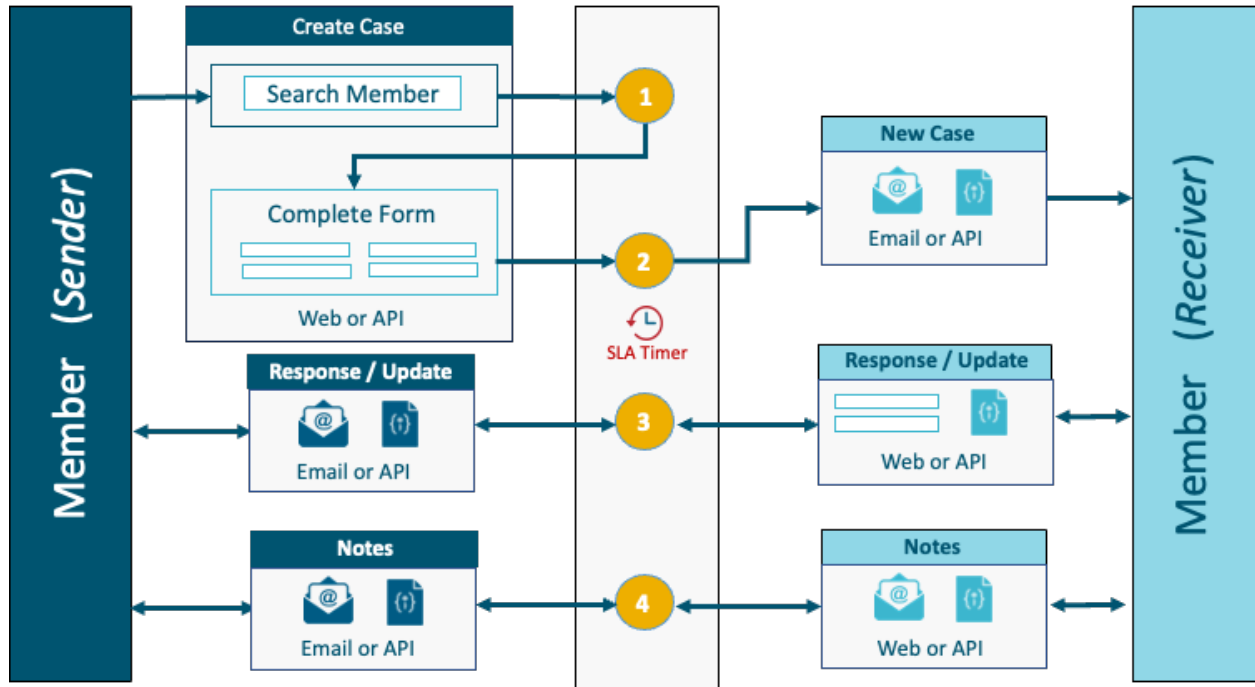
TSANet Member Benefits



TSANet Technology Partner Framework

Benefits of TSANet Connect Integration

Members use the TSANet Connect system to connect and collaborate with other members. The collaboration process is described in the image below:



Collaboration Process – Connect 2.0

Improved User Experience (Integrated B2B)

TSANet Elite Members can integrate their system into TSANet Connect to improve the user experience and process compliance. User Experience benefits are noted below.

Use Case	WebApp Collaboration	Integrated B2B
Inbound request from a member	Uses an alias or email-to-case feature of the target system.	Create a case in the system with the ability to utilize all features of the system for automation and routing
Outbound request to a member	Fragmented process. Member System, TSANet WebApp and engineer's email	End-End Process is within the Members system and workflow.
Ongoing notes and updates	user works in email/notes and manually updates the case	Updates added to the case as part of TSANet Connect notes and update API

Note: An example of user experience can be found in this document's Salesforce reference integration section.

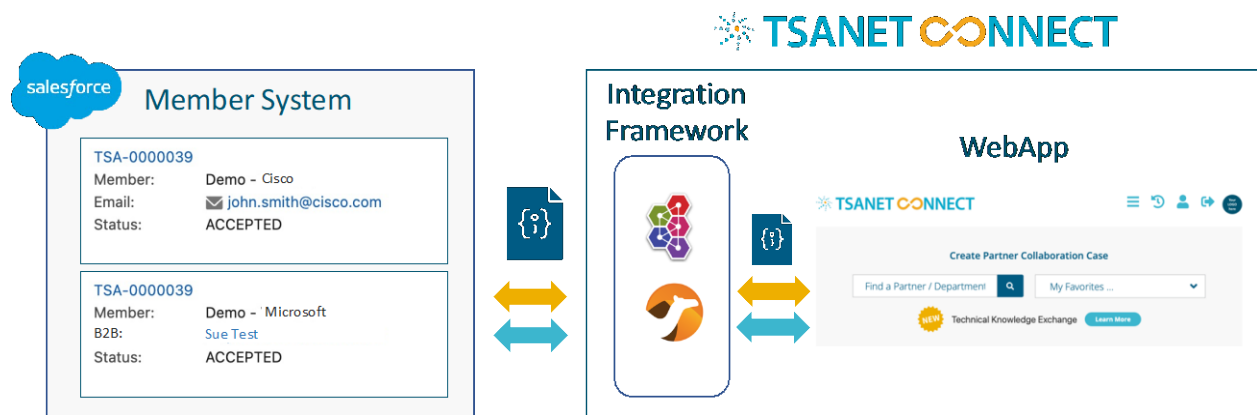
Benefits vs. Point-Point integrations

The table below highlights some of the challenges with Point-Point integrations and the benefits of TSANet Connect integration for the use cases described in this document.

Topic	Point – to - Point	TSANet Connect
Implementation	Multi-company IT project. Increases approvals, complexity, and cost. Avg \$100k to put a point – point connection in place. Multiply this by number of Partners	Each Member has a single integration project with TSANet Connect. This then enables that Member to collaborate with all other 900+ members
Ongoing support / Change management	Change management must be coordinated between the vendors. This increases the cost of ongoing support	TSANet manages changes (Example adding a field that they require for a new case). Members see this change the next time they make a collaboration request
Flexibility to support all Partners	Can only work with Partners that have similar IT capabilities. Most small and medium-sized companies do not have the ability to integrate systems.	TSANet members can use the TSANet Connect WebApp or use the APIs to fully integrate their systems. Both Standard Email collaboration and B2B collaborations are supported from the same process.

System Overview

The TSANet Connect system includes a Web App and an Integration Framework based on Apache Camel on Spring Boot. The Integration Framework provides a complete set of REST APIs for creating integrations. This document will describe the TSANet Connect REST APIs and how they are used to create the Salesforce Reference Implementation.



TSANet Connect – Integration Framework

The TSANet Connect Integration Framework is based on Apache Camel on Spring Boot. Apache Camel is an open-source integration framework facilitating seamless communication between applications, systems, and services. It simplifies connecting disparate components by providing a flexible and extensible platform. In addition to the provided TSANet Connect REST APIs, the system can support custom integrations as needed.

Key features of Apache Camel:

- **Routing Engine:** Camel allows you to define routes that connect various endpoints (such as databases, messaging systems, APIs, and files). These routes guide the flow of data between components.
- **Component-Based:** Camel supports many components (e.g., HTTP, JMS, FTP, REST) out of the box. You can also create custom components tailored to your specific needs.
- **Domain-Specific Language (DSL):** Camel offers a concise DSL for defining integration routes. This DSL abstracts away the complexities of low-level APIs.

- Enterprise Integration Patterns (EIP): Camel implements EIPs, which provide standardized solutions for common integration challenges.
- Extensibility: You can easily extend Camel by adding custom processors, components, and data transformations.
- Testing Support: Camel provides robust testing capabilities to ensure the correctness of your integration flows.

TSANet Connect – Database Structure

TSANet Connect is designed to provide a simple and flexible method for Members to collaborate. The following objects are found in the database to support the collaboration process:

Groups: Groups are used to create relationships between Members. A Member could be part of several groups. One-Many and Many-Many groups are supported. TSANet-hosted groups and Member-hosted groups are used. Groups define the rules for collaboration (For example SLA response times)

Accounts: Accounts provide high-level information about a Member.

Departments: An account can have one or more departments. This allows you to assign separate process forms for departments (e.g., *Division A and Division B*).

Users: Users are associated with an account. A Members Identify management system can manage user access through SAML2.0.

Process Forms: An account can have one or more forms. These forms are linked to a department or group to support inbound workflow. They define what data the member receives for inbound requests.

TSANet Cases: Metadata for a collaboration request (Sender and Receiver metadata).

- **Case Notes:** Notes related to a collaboration

A facade pattern has been implemented to facilitate the translation of API consumer language to database persistence language.

TSANet Connect 2.0 REST API

The following list of available APIs is in OSA3.0 format:

TSANet provides access to the TSANet Connect Beta environment for member integration and testing. Full API documentation is provided in **Swagger**, and a **Postman** collection is also provided. The following sections provide an overview of the API and its use. Contact TSANet at membership@tsanet.org to request access to the Beta site. The Salesforce reference integration section provides an example of how these APIs are used.

Endpoint (POST): login

Use: Login to TSANet and get a token for further calls.

```
{
  "username": "string",
  "password": "string"
}
```

Output

```
{
  "accessToken": "string",
  "tokenType": "string",
  "expiresIn": 0
}
```

Endpoint (GET): me

Use: Returns the TSANet user profile.

```
{
  "company": {
    "id": 0,
    "name": "string",
    "description": "string"
  },
  "user": {
    "id": 0,
    "username": "string",
    "firstName": "string",
    "lastName": "string",
    "email": "string",
    "phone": "string",
    "phoneCountryCode": "string",
    "city": "string"
  },
  "encodedCredentials": "string"
}
```

Endpoint (GET): partners (search)

Use: Search for a partner or department. This searches member and department names to return a companyId and departmentId if the member has multiple departments.

```
[
  {
    "label": "string",
    "companyId": 0,
    "departmentId": 0
  }
]
```

Endpoint (GET): form/department

Use: Get the collaboration form for a specific department in a member's company.

```
{
  "documentId": 0,
  "internalCaseNumber": "string",
  "optionalReceiverInternalCaseNumber": "string",
  "problemSummary": "string",
  "problemDescription": "string",
  "casePriority": "string",
  "readonlyAdminNote": "string",
  "readonlyEscalationInstructions": "string",
  "testSubmission": true,
  "customerData": [
    {
      "id": 0,
      "section": "string",
      "fieldName": "string",
      "value": "string"
    }
  ]
}
```

Endpoint (GET): form/company

Use: Get the collaboration form for a member company without departments setup in TSANet.

```
{
  "documentId": 0,
  "internalCaseNumber": "string",
  "optionalReceiverInternalCaseNumber": "string",
  "problemSummary": "string",
  "problemDescription": "string",
  "casePriority": "string",
  "readonlyAdminNote": "string",
  "readonlyEscalationInstructions": "string",
  "testSubmission": true,
  "customerData": [
    {
      "id": 0,
      "section": "string",
      "fieldName": "string",
      "value": "string"
    }
  ]
}
```

Endpoint (GET): form/documentID/field

Use: Get custom fields for a form

```
{
  "documentId": 0,
  "fieldId": 0,
  "section": "string",
  "label": "string",
  "options": "string",
  "additionalSettings": "string",
  "type": "string",
  "displayOrder": 0,
  "validationRules": "string",
  "value": "string",
  "required": true
}
```

Endpoint (GET): form/documentID/field/selections

Use: Get values for field types “Select” and Tier Select”

```
[
  {
    "value": "string",
    "children": [
      "string"
    ]
  }
]
```

Endpoint (POST): cases/create

Use: Create TSANet collaboration request case by submitting the form

```
{
  "documentId": 0,
  "internalCaseNumber": "string",
  "optionalReceiverInternalCaseNumber": "string",
  "problemSummary": "string",
  "problemDescription": "string",
  "casePriority": "string",
  "readonlyAdminNote": "string",
  "readonlyEscalationInstructions": "string",
  "testSubmission": true,
  "customerData": [
    {
      "id": 0,
      "section": "string",
      "fieldName": "string",
      "value": "string"
    }
  ]
}
```

Output (Summary)

```
{
  "id": 0,
  "submitCompanyName": "string",
  "submitCompanyId": 0,
  "submitterCaseNumber": "string",
  "receiveCompanyName": "string",
  "receiveCompanyId": 0,
  "receiverCaseNumber": "string",
  "summary": "string",
  "description": "string",
  "priority": "string",
  "status": "string",
  "token": "string",
  "createdAt": "2024-06-14T21:0
```

Endpoint (GET): cases/(InternalCaseNumber)

Use: Get a list of cases by ID (member's CRM case ID)

```
[
  {
    "id": 0,
    "submitCompanyName": "string",
    "submitCompanyId": 0,
    "submitterCaseNumber": "string",
    "receiveCompanyName": "string",
    "receiveCompanyId": 0,
    "receiverCaseNumber": "string",
    "summary": "string",
    "description": "string",
    "priority": "string",
    "status": "string",
    "token": "string",
    "createdAt": "2024-06-14T21:29:33.217Z",
    "updatedAt": "2024-06-14T21:29:33.217Z"
  }
]
```

Endpoint (GET): cases

Use: Get a list of cases. Allows filter by (Example inbound, Open)

```
[
  {
    "id": 0,
    "submitCompanyName": "string",
    "submitCompanyId": 0,
    "submitterCaseNumber": "string",
    "receiveCompanyName": "string",
    "receiveCompanyId": 0,
    "receiverCaseNumber": "string",
    "summary": "string",
    "description": "string",
    "priority": "string",
    "status": "string",
    "token": "string",
    "createdAt": "2024-06-14T21:25:39.823Z",
    "updatedAt": "2024-06-14T21:25:39.823Z"
  }
]
```

Endpoint (POST): case/approve

Use: Respond to a collaboration request case

```
{
  "caseNumber": "string",
  "engineerName": "string",
  "engineerPhone": "string",
  "engineerEmail": "string",
  "nextSteps": "string"
}
```

Output

```
{
  "id": 0,
  "submitCompanyName": "string",
  "submitCompanyId": 0,
  "submitterCaseNumber": "string",
  "receiveCompanyName": "string",
  "receiveCompanyId": 0,
  "receiverCaseNumber": "string",
  "summary": "string",
  "description": "string",
  "priority": "string",
  "status": "string",
  "token": "string",
  "createdAt": "2024-06-14T21:31:14.905Z",
  "updatedAt": "2024-06-14T21:31:14.905Z",
}
```

Endpoint (POST): case/update/approval

Use: Update a case response (Example assigned engineer change)

```
{
  "caseNumber": "string",
  "engineerName": "string",
  "engineerPhone": "string",
  "engineerEmail": "string",
  "nextSteps": "string"
}
```

Output (same as approve)

Endpoint (POST): cases/notes/create

Use: Post Notes to a case.

```
{
  "summary": "string",
  "description": "string",
  "priority": "string"
}
```

Output

```
[
  {
    "id": 0,
    "caseId": 0,
    "creatorUsername": "string",
    "summary": "string",
    "description": "string",
    "priority": "string",
    "status": "string",
    "token": "string",
    "createdAt": "2024-06-14T21:
36:53.859Z",
    "updatedAt": "2024-06-14T21:
36:53.859Z"
  }
]
```

Endpoint (GET): cases/notes

Use: Get Notes for a case

```
[
  {
    "id": 0,
    "caseId": 0,
    "creatorUsername": "string",
    "summary": "string",
    "description": "string",
    "priority": "string",
    "status": "string",
    "token": "string",
    "createdAt": "2024-06-14T21:
38:07.303Z",
    "updatedAt": "2024-06-14T21:
38:07.303Z"
  }
]
```

Integration Process

Beta Environment

Elite Members can access the Beta environment to develop and test integrations. TSANet will provide the following:

1. User account on the system (Beta-Member_Name). User/password for API
2. Connection to the Test Account or other Members also in Beta to support end-end testing.
3. Link to Swagger documents
4. Postman collection
5. Access to the Salesforce Package for Members using Salesforce.
6. Access to a TSANet developer resource for questions and assistance with integration.

TSANet will work with members to co-develop standard integrations for reuse in the most common systems. For an example of this approach, see the Salesforce reference implementation section.

Production Environment

After validating the Member integration, TSANet will work with the Member to transition to the Production environment.

Reference Integration – Salesforce

TSANet provides a Salesforce Package as a reference integration to show the steps to building integration and the user experience. TSANet will provide the unlocked Salesforce package with full access to the source code. When enhancements are made in the unmanaged package, TSANet will review them with the member to determine if the improved functionality should be part of the reference package.

View the Webinar below for an overview and demo of the Salesforce integration:

<https://www.tsanet.org/tsanet-connect-update-webinar-may-7-2024/>



Salesforce Install Guide

The Salesforce install guide provides the information needed to install and configure the Salesforce package. Members will receive this document as part of onboarding to the Beta environment.

Salesforce package developer guide

The Salesforce package developer guide includes details on accessing the package's source code and making changes. This document will be provided to Members who wish to enhance the Unlocked Salesforce package.

Salesforce Package Components

After installation, the TSANet Connect Package adds the following to Salesforce.

TSANet Case Object and Child Objects

The TSANet Case Object holds data sent to and from the TSANet Connect system. Child records provide case responses and notes during the collaboration's lifecycle. Each TSANet Collaboration case is linked to a case in the member's system.

The screenshot displays the TSANet Case Object interface for case TSA-0000024. The interface is divided into several sections:

- Header:** Shows the case ID "TSA-0000024" and a "+ Follow" button.
- Details:** A table of case information:

TSANet Case	TSANet Caseld
TSA-0000024	2530
Case	Status
00001036	ACCEPTED
Priority Note	Priority
Initial response within 2 hours . The Mutual Customer's entitlement with member may affect response times	HIGH
Type	Request Date
Inbound	6/3/2024, 8:22 PM
Summary	Response SLA
inbound case	6/3/2024, 10:22 PM
Description	Token
inbound case	KyIgbntTIX8eMEYLhdX
- Member Information:** A table showing member details:

Company Name (Submitted)	Company Name (Received)
Test Company 2	Test Company 1
Name	Engineer Name
- TSANet Responses (1):** A list of responses, including one with ID "TSA-R-000009" and note "Our assigned engineer will contact you." A "View All" link is present.
- TSANet Notes (1):** A list of notes, including one with ID "N-00014" and description "outbound note". A "View All" link is present.

TSANet Case and Child Objects

TSANet LWC

A lightning web component provides the functionality for making an outbound request. This component is added to the case or could be exposed as part of a workflow trigger. The user can mouse over the TSANet case to view additional details, and all updates and notes are integrated into the case feed.



The screenshot shows the TSANet Connect LWC (Lightning Web Component) for case TSA-0000039. It displays the following information:


- Header:** "TSANet Cases (1)" with a dropdown arrow.
- Case ID:** TSA-0000039
- Member:** Demo - Cisco
- Email:** john.smith@cisco.com
- Status:** ACCEPTED

TSANet Connect LWC

Case Feed Updates


The package provides a method to update the case feed and post notes.



✓  **Mister Manager** June 3, 2024 at 1:57 PM 


 To: Internal

Your collaboration request to Test Company 2 for Test Company 1 Case# 00001036 has been sent. You should receive a response within 4 hours. If you need to escalate this case follow the instructions below:

Request Details:
Summary: Outbound case 3:57pm
Description: Outbound case 3:57pm
Priority: LOW
status: OPEN
createdAt: 2024-06-03 20:57:37
firstName: Mister
lastName: Manager
email: manager@companyone.com
phone: 4155551212
phoneCountryCode: VG
city: test

 Comment

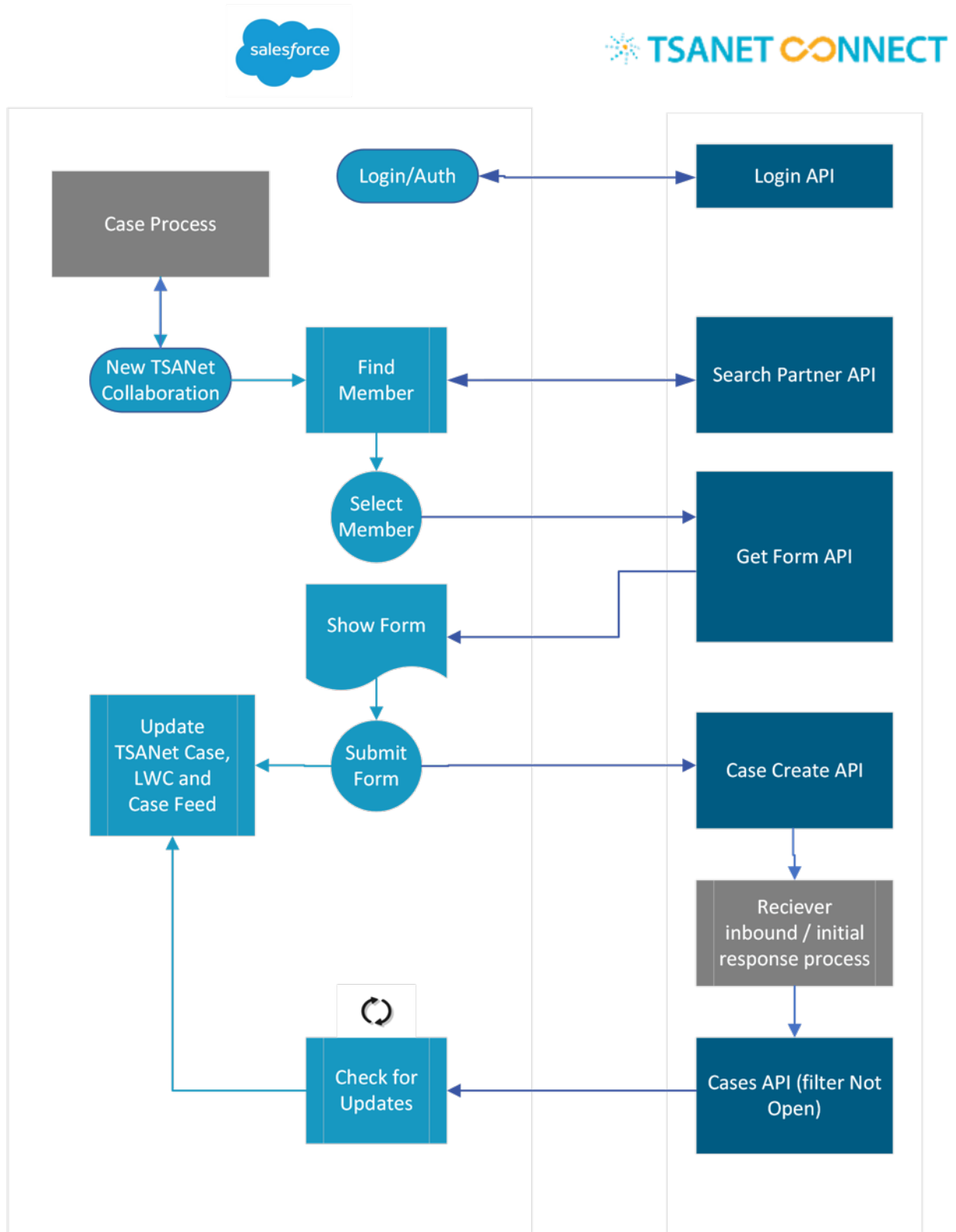
✓  **Mister Manager** June 3, 2024 at 1:41 PM 

 To: Internal

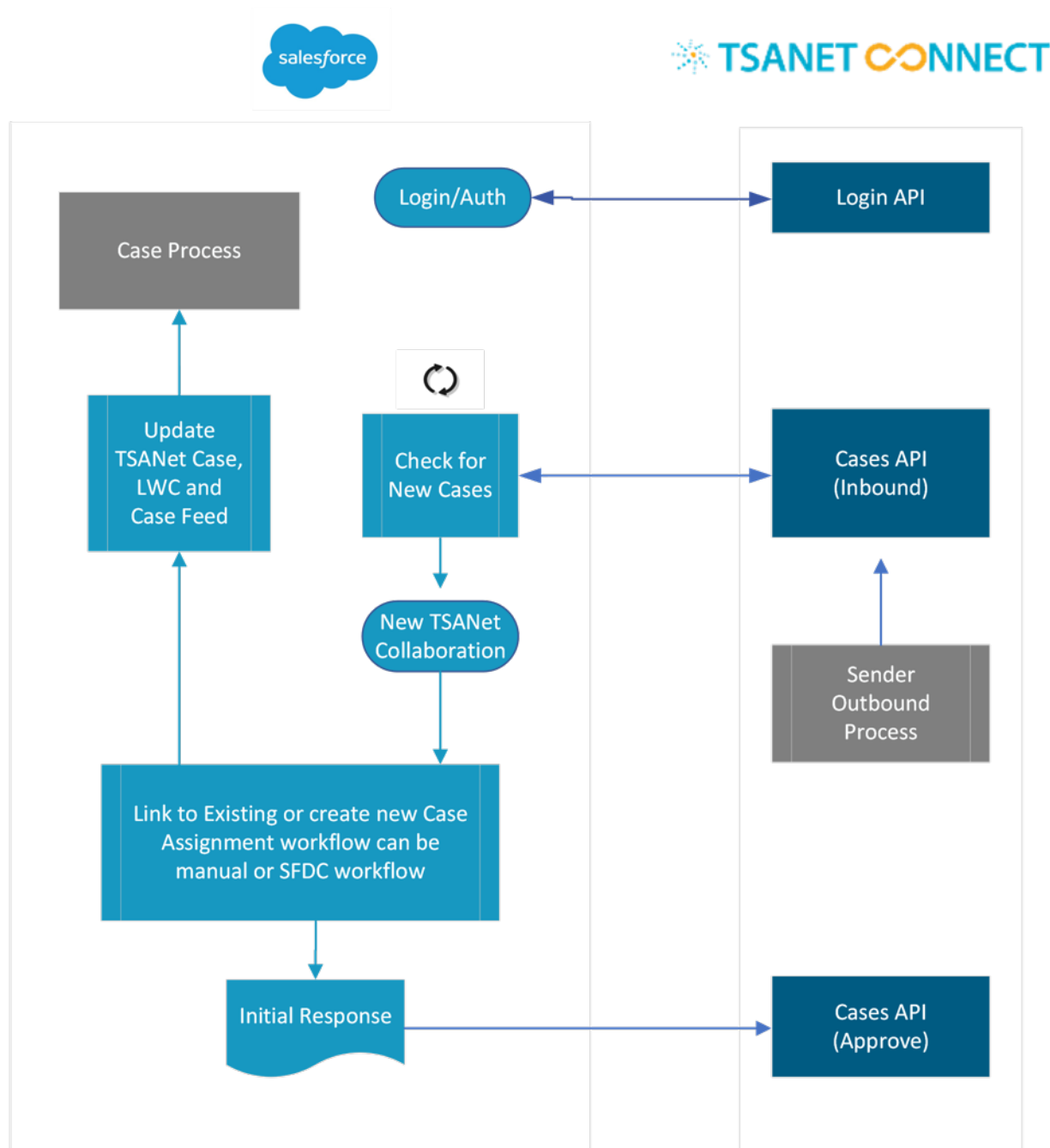
Note Details:
Summary: outbound note 2
Description: outbound note 2

Feed is updated for all activity on a TSANet Case

User Experience – Outbound Request (To Member)



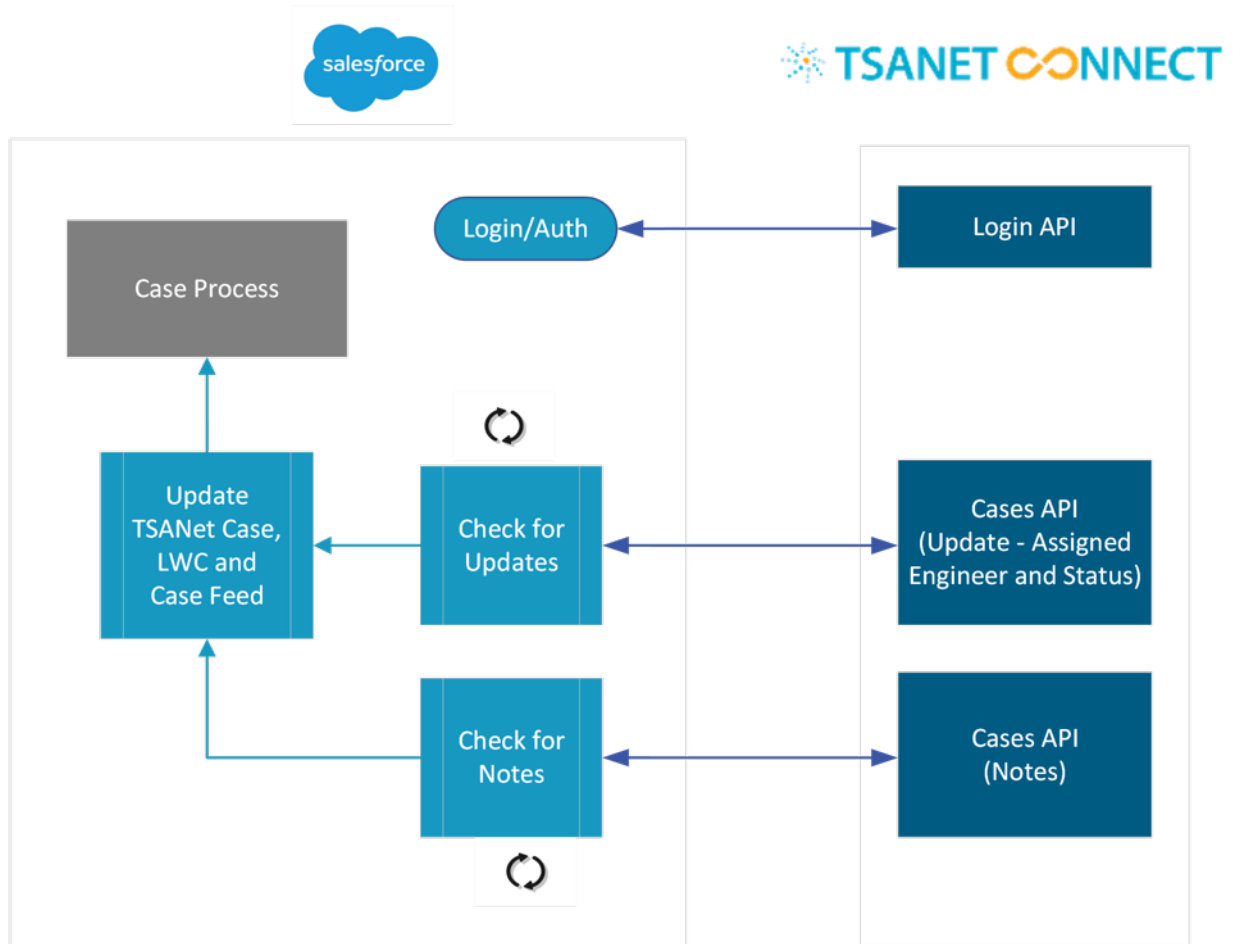
User Experience – Inbound Request (From Member)



Note: The package provides data and screens for the manual process. Members can implement workflow to automate the inbound process as needed.

Ongoing Updates – Collaboration with Connect 2.0 Members.

When collaborating with other TSANet Members, the system supports bi-directional notes in the case feed. Initial response updates are also supported.



Ongoing Updates – Response and Notes Feature.

Members can control the information they send to collaborating Members. The best practice is to allow the user to define an update's visibility. TSANet suggests these options:

- Customer + TSANet Member: updates both customer and collaborating member
- TSANet Member: Only include the TSANet member
- Internal Only: for internal notes

Response updates support updating case information, such as the assigned engineer.

TSANet Case Status and Priority.

The TSANet case status and Priority are separate from the member's case. Members can map these to their case or keep them separate.

Priority: P1 (High), P2 (Medium), P3 (Low). The system defines and monitors each SLA's initial response. If the SLA is not met, an escalation workflow triggers, alerting both the sender and receiver management contacts. Default SLAs are set in TSANet Hosted groups (P1=2hr, P2=4hr, P3=24Hr). Member can modify these defaults in their Hosted groups.

Status: The following status definitions are used.

- **Open** (New waiting initial response)
- **Information** (Sent if information is needed to accept or reject)
- **Accepted** (Collaboration Accepted)
- **Rejected** (Collaboration rejected. For example, end customer does not have support). This status is not used for solution support.
- **Closed** (set by submitter or auto-close after 30 days of inactivity)