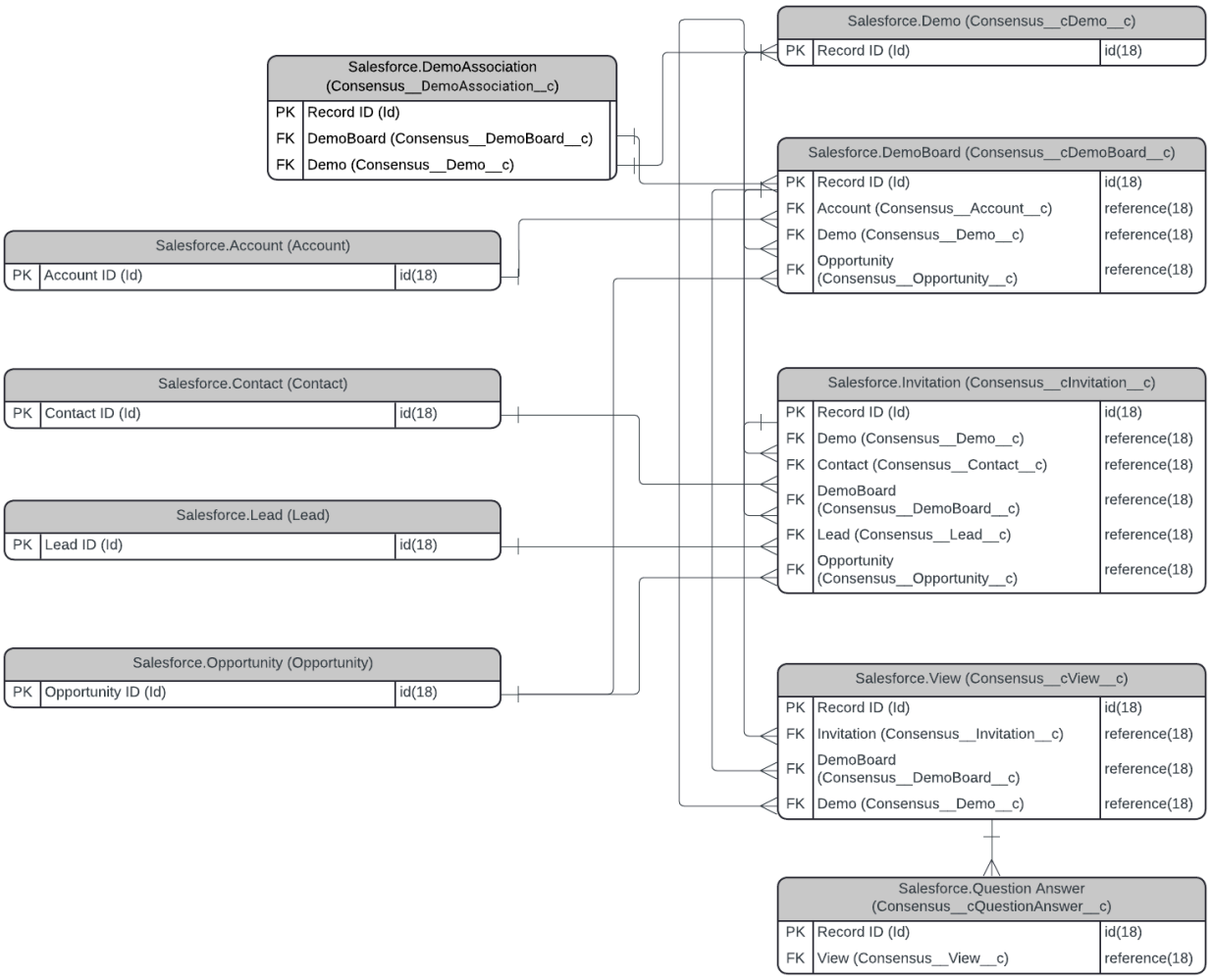
SFDC 11.13.0 Technical Guide (+ Saleo)

- Data schema (with explanation of fields)
  - List custom Objects, Fields and Flow automations
    - Standard Objects
      - Account (Saleo fields)
      - Contact
      - Lead
      - Activity
    - Custom Objects
      - Demo
      - Demo Association
      - DemoBoard
      - Invitation
      - View
      - Feature
      - Question Answer
- Apex Scheduled Jobs
  - DemosUpdateBatch
  - DemoRecordsBatch
  - SalesLeadsUpdateBatch
  - MarketingLeadsUpdateBatch
  - ViewsUpdateBatch
  - DemoBoardCrmSyncBatch
    - Contact Object
    - Lead Object
    - Invitation Object (Custom)
    - DemoBoard Object (custom)
    - Demo Association Object (custom)
    - View Object (custom)
    - Feature Object (custom)
    - Task Object
    - Opportunity Contact Role Object
  - RecentDemosUpdateBatch
  - RecentDemoBoards
- Trigger Automations
- Permission Sets & Permission Set Groups
  - Permission Set Groups
  - Permission Sets
    - Consensus Users
    - Consensus Admin
    - Consensus Integration
- Authorization Flows
- Methods of synchronizing data
  - Run Data Update button
  - Schedulers
  - Webhooks
- Create DemoBoard from Consensus Component within Salesforce
- CRM Sync
- Opportunity Recommendations
- Report Types
  - Reports
  - Dashboards

- Data Flow Diagrams
  - Run Date Update
  - Delete Demos, which were deleted in Consensus.
  - Contact Creation
  - Create Lead
  - Contact Role Creation
  - Create Lead by Webhook
- Saleo Integration for Consensus
  - Overview
  - Data schema (with explanation of fields)
    - Standard Objects
      - Account
    - Custom Objects
  - Permission Sets
  - Authentication
    - Required Scopes
    - Callback URLs
  - API Endpoints
    - Authentication
    - Organization
    - Object metadata
    - User queries
    - Account queries
    - Opportunity queries
    - Package detection
    - Flag write
  - Data Flow
    - Flag write on session completion
    - Revenue analytics calculation
  - Security Considerations
    - Data location
    - Encryption
    - Data retention
    - Access control

Data schema (with explanation of fields)



List custom Objects, Fields and Flow automations

Salesforce instance:

Standard Objects

Account (Saleo fields)

API Name	Type
Saleo_Influenced__c	Boolean
Saleo_Agent_Influenced__c	Boolean

Contact

API Name	Type	Value
CreatedByConsensus__c	Boolean	TRUE if created by Consensus
existInConsensus__c	Boolean	TRUE if exists in Consensus

Lead

API Name	Type	Value
Uuid__c	Text	Contains the Consensus recipient ID
existInConsensus__c	Boolean	TRUE if exists in Consensus

Activity

API Name	Type	Value
ExtID__c	Text	External Id to prevent duplication

Custom Objects

Demo

AvgShares__c	Formula
AvgViews__c	Formula
AvgWatchTime__c	Text
DaysSinceLastActivityDemo__c	Formula
LastActivityDate__c	Roll-up Summary
Total_Watch_Time_HMS__c	Text
TotalDemoBoards__c	Number
TotalShares__c	Roll-up Summary
TotalViews__c	Roll-up Summary
TotalWatchTime__c	Roll-up Summary

Demo Association

Demo association object is a junction object between DemoBoard and Demo, and it is used to handle the MDS for Salesforce.

Demo__c	Lookup(Demo)	
DemoBoard__c	Lookup(DemoBoard)	
DemoBoardName__c	Formula	The name of the related demoboard
ExtId__c	Text	External Id to prevent duplication
Name	Text	The name of the related demo

DemoBoard

Account__c	Lookup(Account)
AvgDBShares__c	Formula
AvgDBViews__c	Formula
AvgDBWatchTime__c	Formula
CreatedById	Lookup(User)
CreationDate__c	Date/Time
CreationDateFormula__c	Formula
CrmSyncAttempts__c	Number
CrmSyncFlaggedAt__c	Date/Time
CrmSyncPending__c	Boolean
DaysSinceLastActivity__c	Formula
Demo__c	Lookup(Demo)
DemoBoardName__c	Text
DemoBoardOwner__c	Text
DemoBoardUUID__c	Text
DetailsURL__c	URL
ExtID__c	Text

HotspotViews__c	Number
InitialOpportunityStage__c	Text
LastDemoActivityDate__c	Date/Time
LastModifiedById	Lookup(User)
Lead__c	Lookup(Lead)
MarkedAsTest__c	Boolean
Name	Auto Number
Opportunity__c	Lookup(Opportunity)
OpportunityStage__c	Text
Organization_Name__c	Text
OwnerId	Lookup(User,Group)
Subtitle__c	Text
TotalInvitations__c	Number
TotalShares__c	Number
TotalViews__c	Number
TotalWatchTime__c	Number
TotalWatchTimeHMS__c	Formula
TourViews__c	Number

Invitation

AssignedToOpp__c	Boolean	
Contact__c	Lookup(Contact)	
ContactSource__c	Text	
ConvertedOpportunity__c	Lookup(Opportunity)	
CreationDate__c	Date/Time	
CreationDateFormula__c	Formula	

DaysSinceLastActivity__c	Number	
Demo__c	Master-Detail(Demo)	
DemoBoard__c	Lookup(DemoBoard)	
DemoBoardOwner__c	Text	
Email__c	Email	
ExtId__c	Text	
Hash__c	Text	
HeatmapURLV11__c	Formula	
InternalUser__c	Boolean	
Lead__c	Lookup(Lead)	
Mode__c	Text	
Opportunity__c	Lookup(Opportunity)	
OpportunityStage__c	Text	
OwnerUuid__c	Text	
Person__c	Formula	
Primary__c	Text	
RecipientUuid__c	Text	
SendDemoHash__c	Text	
SendDemoUuid__c	Text	
SharedByName__c	Text	
Shares__c	Number	
TimeLastView__c	Date/Time	
TotalViewTime__c	Number	
TotalWatchTimeHMS__c	Formula	
Type__c	Text	
UserId__c	Text	

Uuid__c	Text	(Text)
Views__c	Number	

View

Demo__c	Lookup(Demo)
DemoBoard__c	Lookup(DemoBoard)
Duration__c	Number
Duration__c	Number
DurationHMS__c	Long Text Area
Files__c	Number
Invitation__c	Master-Detail(Invitation)
linkUuid__c	Text
Uuid__c	Text
ViewDate__c	Date/Time
ViewId__c	Text

Feature

ExtId__c	Text
Title__c	Text
Value__c	Text
View__c	Lookup(View)

Question Answer

View__c	Master-Detail(View)
Question__c	Text

Answer__c	Text
GlobalActionId__c	Text
ExtId__c	Text

Apex Scheduled Jobs

Batch job size can be checked with `Integer PackageAdmin.getBatchSize(String className)`
Batch jobs size can be changed to a value between 10 and 500 using `void PackageAdmin.setBatchSize(String className, Integer size)`. Note: Changing the batch size is inadvisable without confirming with Consensus.

DemosUpdateBatch

Job type: Batch

Interval: Every 24 hours, or manually if admin selects “Run Data Update.”

Description: upsert Demo records

API endpoints: 'api/integr/v1.0/demo/search' (all Demos)

Default batch sizes for each request: 200

Which field updating in SFDC (Demo__c)	**Which parameters we use from response**
ExternalTitle__c	internalTitle
Name	
Type__c	type
uuid__c	uuid
uuidType__c	uuidType
isPublic__c	isPublic
isArchived__c	isArchived
isPublished__c	isPublished
isActive__c	
UserID__c	
Mode__c	
ExtId__c	

PermissionUUID__c	permissions.uuid/folderPermissions.uuid
	createdAt
	description
	folderPermissions.value
	folderPermissions.folderName

DemoRecordsBatch

Job type: Batch

Interval: Every 24 hours, or manually if admin selects “Run Data Update.”

Description: deletes inactive demos. If an inactive demo becomes active again it will update the Demo record.

1. if Demo__c.isActive__c is TRUE, update Demo records.
2. if Demo__c.isActive__c is FALSE, delete Demo records

SalesLeadsUpdateBatch

Job type: Batch

Interval: Every 24 hours, or manually if admin selects “Run Data Update.”

Description:

1. create Contact if externalOpportunityId parameter not empty AND Consensus__c.isCreateContacts__c is checked;
2. create Lead if Consensus__c.isLoadHistoricalActivitySch__c is checked OR type parameter is Sales AND isCreateRecipientsShedul__c is checked;
3. upsert DemoBoard__c records;
4. upsert cInvitation__c records;
5. Upsert ContactRoles if Consensus__c.isCreateContactsRoles__c is checked

API endpoints: 'api/integr/v1.0/report/trackDemoboards'

batch sizes for each request: 100

MarketingLeadsUpdateBatch

Job type: Batch

Interval: Every 24 hours, or manually if admin selects “Run Data Update.”

Description:

1. creates Contact if externalOpportunityId parameter not empty AND Consensus.isCreateContacts__c is checked AND Consensus.isCreateLeadsSchedul__c is not checked;
2. creates Lead if Consensus__c.isLoadHistoricalActivitySch__c is checked OR type parameter is Marketing AND isCreateLeadsSchedul__c is checked;
3. upsert DemoBoard__c records;
4. upsert cInvitation__c records;
5. Upsert ContactRoles if Consensus__c.isCreateContactsRoles__c is checked

API endpoints: 'api/integr/v1.0/marketing/searchleads'

batch sizes for each request: 150

ViewsUpdateBatch

Job type: Batch

Interval: Runs at the end of every recent data sync (every Consensus__c.schedulerTimeInterval__c minutes, default 10) and at the end of the daily full data sync. It doesn't start if a run is already in progress or one finished in the last 5 minutes. Admins can also start it manually: PackageAdmin.resetViewsSync(fromDateTime) resyncs views from the given date and restarts the job right away.

Description:

Fetches views that occurred since the last successful run. The start point is Consensus__c.ViewsLastSyncedTo__c. If that's empty, it uses Consensus__c.StartSyncDate__c, and if that's empty too, it uses 01/01/2000. Each run reads pages until it reaches the current time, saving its position in Consensus__c.ViewsLastUuid__c, so a failed run picks up where it stopped.

1. Upsert cView__c records (external ID Uuid__c). It only does this for views that match an existing cInvitation__c (same DemoBoard and recipient); views with no match are skipped and logged.
2. upsert cFeature__c records for each view.
3. upsert QuestionAnswer__c records for each view.
4. upsert Task ("Demo watched") on the Invitation's Contact or Lead if Consensus__c.EnableTaskCreation__c is checked.
5. update Consensus__c.ViewsLastSyncedTo__c / ViewsLastUuid__c when the run finishes.

API endpoints: 'api/v2/views' (GET, OAuth Bearer token)

batch sizes for each request: 100 (configurable via Consensus__c.ViewsUpdateBatchSize__c)

DemoBoardCrmSyncBatch

Job type: Batch

Interval: Runs at the end of every recent data sync (every Consensus__c.schedulerTimeInterval__c minutes, default 5) and at the end of the daily full data sync, but only if some DemoBoards are flagged for sync (CrmSyncPending__c = true).

Description:

Sends the current Opportunity and Account links of flagged DemoBoards to Consensus. Right now DemoBoards only get flagged when an Account merge moves them to a different Account, which happens without a DemoBoard trigger firing. The Account comes from the DemoBoard's Opportunity, or from cDemoBoard__c.Account__c when there is no Opportunity;

update cDemoBoard__c records: on success, it clears CrmSyncPending__c, CrmSyncAttempts__c and CrmSyncFlaggedAt__c. On failure, it adds 1 to CrmSyncAttempts__c and the DemoBoard stays flagged for the next run. DemoBoards that still fail 30 days after being flagged are unflagged and logged;

if a request fails, it skips the rest of the run, and the next run starts with the DemoBoards that have the fewest failed attempts.

API endpoints: 'api/integr/v1.0/link/updateLinkExternalOppAccData'

batch sizes for each request: 10 DemoBoards (the API's maximum). Up to 200 DemoBoards per run, configurable via Consensus__c.CrmSyncBatchSize__c

Objects and fields created/updated from the above Data Sync jobs:

Contact Object

Which field updating in SF (Contact__c)	Which parameters we use from response
Email	email
	externalOpportunityId
	type
firstName/lastName	name

Title	title
AccountId	externalAccountId
CreatedByConsensus__c	

Lead Object

Which field updating in SF (Lead__c)	**Which parameters we use from response**
	creationDate
Email	email
	type
firstName/lastName	name
Company	organization
email	email
Title	title
Uuid__c	contactUuid
Phone	
Country	
State	
Description	
LeadSource	
existInConsensus__c	

Invitation Object (Custom)

Which field updating in SF (cInvitation__c)	**Which parameters we use from response**
Demo__c	demoUuid
CreationDate__c	creationDate
DaysSinceLastActivity__c	timeLastView
Email__c	email

Hash__c	hash
Mode__c	
Primary__c	invites.primary
SendDemoHash__c	senddemoHash
SendDemoUuid__c	senddemoUuid
DemoBoardOwner__c	ownerName
ContactSource__c	invites.sharedByName
AssignedToOpp__c	
Opportunity__c	externalOpportunity.externalOpportunityId
ConvertedOpportunity__c	
OpportunityStage__c	
HeatmapURL__c	
DetailsURL__c	senddemoUuid + ...
Shares__c	shares
TimeLastView__c	timeLastView
TotalViewTime__c	viewTime
Type__c	
UserId__c	
Uuid__c	senddemoUuid+hash
Views__c	invites.views
Extid__c	
InternalUser__c	isTest
Contact__c	
Lead__c	

DemoBoard Object (custom)

Which field updating in SF (cDemoBoard__c)	Which parameters we use from response
Organization_Name__c	Organization
DemoBoardName__c	demoboardName
Demo__c	demoUuid
CreationDate__c	creationDate
DemoBoardUUID__c	senddemoUuid
Opportunity__c	externalOpportunity.externalOpportunityId
Account__c	externalOpportunity.externalAccountId
Extid__c	senddemoUuid + ...
MarkedAsTest__c	isTest
Subtitle__c	opportunity
HotspotViews__c	totalHotspotOpened
TourViews__c	totalTourViews
InitialOpportunityStage__c	

Demo Association Object (custom)

Which field updating in SF (DemoAssociation__c)	Which parameters we use from response
DemoBoards__c	Parent DemoBoard Id
Demo__c	Related Demo Id
Name	Demo Name
ExtId__c	demoboard.DemoBoardUUID__c + '_' + demo.uuid__c

View Object (custom)

Which field updating in SF (cView__c)	Which parameters we use from response
Demo__c	invites.viewData.demoUuid

Duration__c	invites.viewData.duration
Files__c	invites.viewData.files
Replay__c	invites.viewData.isReplay
linkUuid__c	invites.viewData.linkUuid
ViewId__c	invites.viewData.viewId
ViewDate__c	invites.viewData.dateView
selectedDemoUuid__c	invites.viewData.selectedDemoUuid
Uuid__c	invites.viewData.viewId + ...
	features
	Viewid
	demoTitle
Invitation__c	

Feature Object (custom)

Which field updating in SF (cFeature__c)	Which parameters we use from response
Name	invites.viewData.features.uuid
Value__c	invites.viewData.features.value
Title__c	invites.viewData.features.title
ExtId__c	invites.viewData.viewId + '-' + invites.viewData.features.uuid + '-' + ...

Task Object

Which field updating in SF (Task)	**Which parameters we use from response**
Priority	
Status	
Subject	.. + viewData.demoTitle + ...

Description	... _ invites.name + invites.email + invites.organization + invites.title + viewData.demoTitle + ...
ActivityDate	invites.timeLastView
OwnerId	
WhoId/WhatId	
ExtId__c	viewData.demoTitle + '#'+ invites.timeLastView + '#' + invites.contactUuid

Opportunity Contact Role Object

Which field updating in SF (OpportunityContactRole)	**Which parameters we use from response**
OpportunityId	
ContactId	
Role	

RecentDemosUpdateBatch

Job type: Batch

Interval: Set on Configuration page

Description: upsert Demo records which was created today

API endpoints: 'api/integr/v1.0/demo/search' (all Demos)

Batch sizes for each request: 500

The same fields as DemosUpdateBatch are updated

RecentDemoBoards

Job type: Batch

Interval: Set on Configuration page

Description:

Filters based on activity of Today's Date. Runs at desired interval set on the configuration page.

1. create Contact if externalOpportunityId parameter not empty AND Consensus__c.isCreateContacts__c is checked;;
2. create Lead if Consensus__c.isLoadHistoricalActivitySch__c is checked OR type parameter is Sales AND isCreateRecipientsShedul__c is checked;
3. upsert DemoBoard__c records;
4. upsert cInvitation__c records;
5. Upsert ContactRoles if Consensus__c.isCreateContactsRoles__c is checked
6. update cDemo__c fields (TotalDemoBoards__c, AvgWatchTime__c, Total_Watch_Time_HMS__c)
7. Update OpportunityStage__c field on cDemoBoard__c records
8. Update OpportunityStage__c field on cInvitation__c records

API endpoints: 'api/integr/v1.0/report/trackDemoboards'

Batch sizes for each request: 100

The same fields as SalesLeadsUpdateBatch are updated + following analytics fields

Analytics fields:

cDemo__c.TotalDemoBoards__c

cDemo.Total_Watch_Time_HMS__c

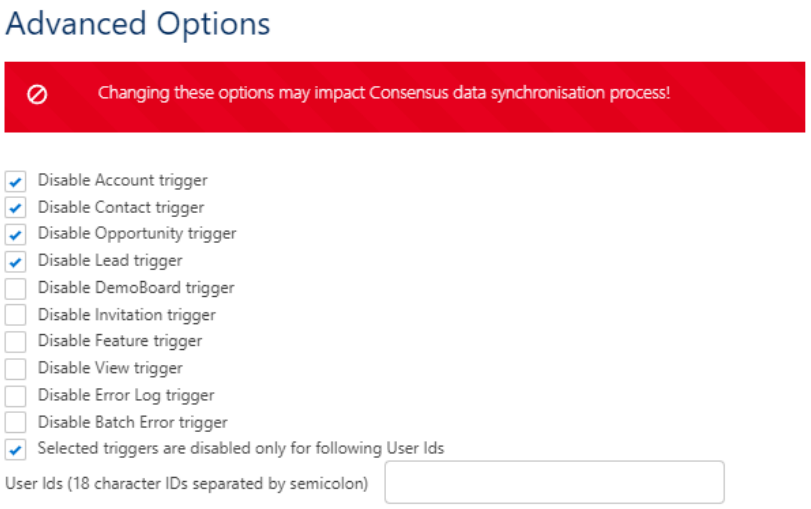
cDemo__c.AvgWatchTime__c

cDemoBoard__c.OpportunityStage__c

cInvitation__cOpportunityStage__c

Trigger Automations

Triggers shouldn’t be disabled without consulting Consensus. All Consensus triggers can be disabled from the Configuration - Advanced Options. Can also be disabled only for a list of users (at most 5)



Account (before delete, after delete)

- On before delete the trigger collects the DemoBoards linked to the Accounts being deleted. These are DemoBoards with a Consensus link (DemoBoardUUID__c filled) that point to the Account directly through the Account field, or through an Opportunity on that Account. The DemoBoards still point to the old Account at this stage, so this is the only point where they can be found.
- On after delete the trigger keeps only the Accounts that were deleted by a merge (Master Record Id is filled) and flags their DemoBoards for CRM sync. By then Salesforce has moved those DemoBoards and Opportunities to the winning Account without firing their own triggers. The flagged DemoBoards are sent to Consensus with their new Account by DemoBoardCrmSyncBatch on the next data sync. If the transaction doesn't have enough room under Salesforce's limits to flag them, DemoBoardCrmSyncQueueable does it instead.
- Ordinary Account deletes (not merges) don't flag anything. The trigger never blocks a merge or delete; all errors are only logged. The trigger can be turned off with the disableAccountTrigger package option (screenshot above).

Contact (before update)

When Contact email is changed related invitations are deleted. Disabled - In some cases Invitations may double depending on the configuration options.

- When a Contact's email is changed or cleared, the trigger deletes the Invitations related to that Contact. This includes Invitations linked to the Contact by a Lead conversion.
- This runs in system mode, so it works for users without the Consensus permission sets. Errors are logged and don't block the Contact update.

Opportunity

This is currently a deprecated trigger. Disabled - No effect

Lead (before update)

When Lead email is changed related invitations are deleted. If Lead is converted Invitations to it are redirected to the created Contact.Disabled - In some cases Invitations may double depending on the configuration options:

- When a Lead is converted, the trigger links the Lead's Invitations to the Contact created by the conversion. It also sets the Invitations' Converted Opportunity field to the Opportunity
  created by the conversion, if there is one. The Invitations stay linked to the Lead as well.
- When a Lead's email is changed or cleared, the trigger deletes the Invitations related to that Lead.
- Both steps run in system mode, so they work for users without the Consensus permission sets. Errors are logged and don't block the Lead update.

DemoBoard (before insert, before update, after insert, after update)

- On before insert the trigger sets the Initial Opportunity Stage field - this field should be set only when the demoboard with opportunity record is created in Salesforce and not changed after that - stops users from editing Initial Opportunity Stage by hand.
- On before update the trigger handles the opportunity change in Consensus and update the Initial Opportunity Stage field if needed.
- On after update the trigger handles the opportunity change in Salesforce and make a callout to Consensus to updated this change and updates the related Invitations to the DemoBoard in Salesforce with the same change.

Invitation (before update, after insert, after update, after delete, after undelete)

- Set value ChildRecordUpdateDate__c field on related Leads to System.now().
- Recalculates rollup fields fields on the DemoBoard object - TotalInvitations__c, TotalViews__c, TotalWatchTime__c:
  - TotalInvitations__c: the number of Invitations
  - TotalViews__c: the sum of Views__c
  - TotalWatchTime__c: the sum of TotalViewTime__

Feature (before update)

When a Feature is updated, the trigger sets the ChildRecordUpdateDate__c field on its parent View to the current date and time. In practice this happens when ViewsUpdateBatch loads Features that already exist in Salesforce; creating a new Feature does not set the field.

- Updating the View also fires the View trigger, which sets ChildRecordUpdateDate__c on the parent Invitation.

- The update runs with the current user's permissions and is skipped when the user can't edit the View's ChildRecordUpdateDate__c field. Errors are logged and don't block the Feature update.
- The trigger is active by default. It can be turned off with the "Disable Feature trigger" option in the Advanced Options on the Configuration page; ChildRecordUpdateDate__c on Views will
  then no longer be updated.

View (before update)

When a View is updated, the trigger sets the ChildRecordUpdateDate__c field on its parent Invitation to the current date and time. In practice this happens when ViewsUpdateBatch loads Views that already exist in Salesforce, or when the Feature trigger stamps the View. Creating a new View does not set the field. Invitations that are being saved in the same transaction are skipped.

- The update runs with the current user's permissions and is skipped when the user can't edit the Invitation's ChildRecordUpdateDate__c field. Errors are logged and don't block the View update.
- The trigger is active by default. It can be turned off with the "Disable View trigger" option in the Advanced Options on the Configuration page; ChildRecordUpdateDate__c on Invitations will then no longer be updated.

Invitation ChildRecordUpdateDate__c is used in one of the Consensus Demolytics dashboards. If customers turn this trigger off, that dashboard column stops updating.

Error Log (after insert)

Creates ErrorLog__c records when Consensus debug mode is active when batch jobs from the package fail. Disabled - ErrorLog__c records wont be created. It:

Runs whenever the package logs an error (an ErrorLogEvent__e platform event). It:

- sends the error messages to Consensus ('api/logger/log') together with the org Id, org name, user Id and package version;
- if "Create Error Logs" is checked on the Configuration page, also creates an ErrorLog__c record for each error, with the exception type, message, stack trace, line number, cause, user and the package class/method. It keeps at most "Error Logs Max Count" records (default 1000) and deletes the oldest ones when the limit is reached.

The trigger can be turned off with the "Disable Error Log trigger" option in the Advanced options on the Configuration page; errors will then be neither sent to Consensus nor saved as ErrorLog__c records.

Permission Sets & Permission Set Groups

Here is a summary of the included permission sets and groups within our managed package. For all integrations to work it is advisable to assign Permission Set groups depending on the user profile, as they contain both custom & standard permissions. Alternatively you can use only permissions sets - Assign ‘Consensus Users’ and ‘Consensus Users (Standard)' for normal users, and ‘Admin Consensus’ and ‘Admin Consensus (Standard)' for Admin users.

Permission Set Groups

1. Consensus Admin Group - Generated on package install. Contains ‘Admin Consensus’ and ‘Admin Consensus (Standard)'
2. Consensus User Group - Generated on package install. Contains ‘Consensus Users’ and ‘Consensus Users (Standard)'
3. Consensus Integration Group - Generated on package install. Contains ‘Integration Consensus’ and ‘Admin Consensus (Standard)'

Permission Sets

1. Consensus Users - Contains permissions for custom objects and custom fields
2. Consensus Users (Standard) - Generated on package install. Contains permissions for standard objects and fields.
3. Admin Consensus - Contains permissions for custom objects and custom fields
4. Admin Consensus (Standard) - Generated on package install. Contains permissions for standard objects and fields.
5. Integration Consensus - Contains permissions for custom objects and custom fields. Designed for Salesforce Integration profile.

Consensus Users

Application

- Consensus (Consensus_Lightning)
- Consensus (Consensus)

Classes

- Activity
- BatchUpdateUserInfo
- cGateway
- cIntegrationService
- cInvitationTriggerHandler
- cInvitationType
- ConfigurationLwcController
- ConsensusSettings
- CreateDemoboardController

- cUtils
- DataUpdateService
- DateUpdateTriggerHandler
- DemoManager
- DemoRecordsBatch
- DemosController (DEPRECATED)
- DemosUpdateBatch
- DemoWebHook
- DemoBoardCrmSyncBatch
- DemoBoardCrmSyncQueueable
- EmailSendController (DEPRECATED)
- ErrorHandler
- Heatmap (DEPRECATED)
- HeatmapController (DEPRECATED)
- IntegrationManager
- IntegrationManager1
- IntegrationManager2
- InvitesComponentController (DEPRECATED)
- LeadContactUpdateHandler
- LoggerService
- MarketingLeadsUpdateBatch
- PackageAdmin
- Permissions
- PostInstallClass
- RecentDemoBoards
- RecentDemosUpdateBatch
- RecentDemosUpdateScheduler
- RecipientsInfo
- Recommendations
- RelatedViewsController
- SearchController (DEPRECATED)
- UpdateOpportunityDataQueueable
- ViewsUpdateBatch
- **Batches, queueables and schedulers**: ViewsUpdateBatch, DemoBoardCrmSyncBatch, DemoBoardCrmSyncQueueable, UpdateOpportunityDataQueueable, SalesLeadsUpdateBatch, SalesLeadUpdateScheduler, MarketingLeadUpdateScheduler, DemoUpdateScheduler, RecentDemoBoardsUpdateScheduler.
- **Trigger handlers**: AccountTriggerHandler, cDemoBoardTriggerHandler.
- **Controllers**: AuraHelperController, ConfigurationController, ConfigurationPageController, FilterController, PageHelperController.

- **Services and helpers**: BitlyService, CheckPermissions, EmailService, QueryHelper, SsoManager, ConsensusSSOPlugin, ViewAllService, UninstallClass.
- **Data types**: Demo, cDemoType, Person, QuestionAnswer, Recipient, ViewType

Custom objects:

Consensus Users Info

- Tab: Visible
- Object: Read
- Field Read Access: All fields
- Field Edit Access: Consensus UUID, selected Email Template

DemoBoards

- Tab: Visible
- Object: Read, Create
- Field Read Access: All fields

Demo Association

- Tab: Hidden
- Object: Read, Create
- Field Read Access: All fields

Demo

- Tab: Visible
- Object: Read
- Field Read Access: All fields

Feature

- Tab: Visible
- Object: Read
- Field Read Access: All fields

Invitation

- Tab: Visible
- Object: Read, Create
- Field Read Access: All fields

View

- Tab: Visible
- Object: Read
- Field Read Access: All fields

Standard objects:

Contact

- Field Edit Access: First Name, Last Name, Title, Email, existInConsensus, Created by Consensus

Lead

- Field Edit Access: Address, ChildRecordUpdateDate, Description, Email, existInConsensus, Lead Source, Loading Historical Data Status, Phone, Title, UUID

Task

- Field Edit Access: Comments, Due Date, ExtID, Name

VisualForce Page

- Demo_Account_Page
- Demo_Contact_Page
- Demo_Lead_Page
- Demo_Opportunity_Page
- Email_Send_Page (DEPRECATED)
- Heatmap_Page (DEPRECATED)

System Permissions

- Access Activities
- Edit Tasks
- View Reports in Public Folders
- View All Custom Settings

Consensus Admin

Consensus Admin permission provides full access to the package metadata. It is designed for an admin users to manage the app. Additionally there some standard permission included on post install script:

Standard objects

Account

- Field Read Access: Industry and Number of Employees

Contact

- Field Edit Access: First Name, Last Name, Title, Email, existInConsensus, Created by Consensus
- Field Read Access: AccountId

Lead

- Field Edit Access: Address, ChildRecordUpdateDate, Description, Email, existInConsensus, Lead Source, Loading Historical Data Status, Phone, Title, UUID
- Field Read Access: Phone, Address

Opportunity

- Field Read Access: Amount, AccountId, Type, Probability

Task

- Field Edit Access: Comments, Due Date, ExtID, Name

System Permissions

- Edit Tasks
- View Reports in Public Folders
- View All Custom Settings
- Access Activities
- View All Users

Consensus Integration

Consensus Integration permission set is cloned from the Consensus Admin and removed the access to all UI elements. It is designed for Salesforce Integration profile which has strong limitation to the Salesforce UI. The permission set has the same permissions as the Consensus Admin except the UI access.

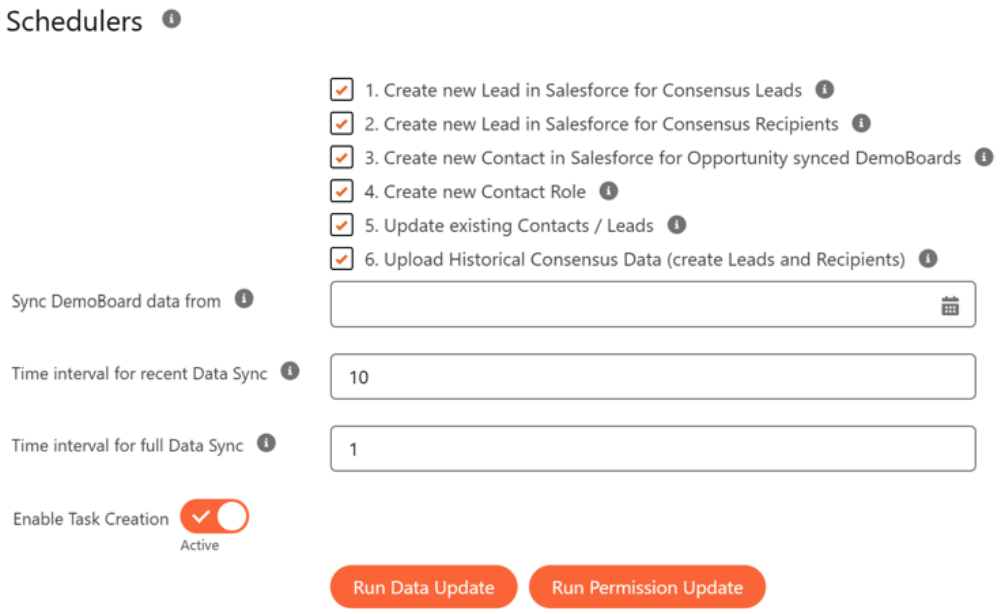
Authorization Flows

The Consensus integration uses two separate OAuth 2.0 connections, one in each direction.

1. **Salesforce → Consensus** (Configuration page "Connect" button). The managed package uses this flow to call the Consensus API (syncing Demos, DemoBoards, Views, recommendations, and logging errors).
   - **Type**: OAuth 2.0 Authorization Code with PKCE (S256).
   - **Setup**: A Consensus admin clicks Connect on the Salesforce Configuration page. The package registers an OAuth client with Consensus automatically, so no API key or secret has to be entered by hand. The admin is then redirected to Consensus to log in and approve access, and comes back to the Configuration page.
   - **Scopes**: user assets.
   - **Tokens**: The refresh token is stored encrypted in the package's protected custom setting. Access tokens are refreshed automatically, and the refresh token is rotated on each refresh. Scheduled jobs and triggers use this connection in the background without the user having to sign in again.
   - **Reconnect**: If the connection stops working (for example the token was revoked or the client is invalid), clicking Reconnect first tries a token refresh. If that fails, it registers a new client and runs the authorization again. The existing credentials are only replaced once the new connection succeeds.

2. **Consensus → Salesforce - CRM Sync flow** (Integration → Salesforce page "Connect" button). Consensus uses this flow to look up Salesforce records when you link a DemoBoard to CRM data in the Consensus web app.
    - **Type**: OAuth 2.0 Authorization Code (web server flow) with PKCE, through the global/shared Consensus Connected App.
    - **Setup**: The Consensus Admin/Integration user authorizes the Consensus Connected App from the Consensus web app with their Salesforce login.
    - **Scopes**: api refresh_token offline_access.
    - **Access**: Read-only. Consensus reads Accounts, Opportunities and Contacts so it can link their IDs to the DemoBoard. It does not create or update Salesforce records.
    - **Policies**: Refresh token rotation is on, and a refresh token expires after 30 days without use. Access follows the user's own Salesforce permissions, and admins can revoke it at any time from Setup → Connected Apps OAuth Usage.

Methods of synchronizing data



Run Data Update button

Order of Operations:

1. Send request to /demo/search for first 500 demos.
2. Receive response.
3. Parse response with received demos.
4. Save demos to Salesforce.
    a. Existing demos will be updated only if there are any updates. (They will not be recreated)

5. Repeat 1-5 steps until the end of the demos list.
6. Send request to /report/trackDemoboards for first 100 demoBoards.
7. Receive response.
8. Parse response with received recipients.
9. Save recipients to Salesforce.
    a. Existing recipients will be updated only if there are any updates. (They will not be recreated)
10. Repeat 6-9 steps till end of Recipients list.
11. **Send request to /marketing/searchleads for first 150 demoBoards.**
12. Receive response.
13. Parse response with received leads.
14. Save leads to Salesforce.
    a. Existing leads will be updated only if there are any updates. (They will not be recreated)
15. Repeat 11-14 steps till end of Leads list.

NOTE: Leads and Recipients from Consensus are both saved as Leads in Salesforce.

Schedulers

Steps:

1. Every X minutes the RecentDemosUpdateBatch apex jobs will fetch data from Consensus
    a. Send request to /demo/search for first 500 demos.
    b. Receive response.
    c. Parse response with received demos.
    d. Save demos to Salesforce.
2. Every X minutes the RecentDemoBoards apex jobs will fetch data from Consensus
    a. Send request to /report/trackDemoboards for the first 100 demoBoards.
    b. Receive response.
    c. Parse response with received recipients.
    d. Save recipients to Salesforce.
3. Every 24 hours send request to /report/trackDemoboards and /marketing/searchleads to true up any date that may have not been synced during the last day.
    a. Send request to /report/trackDemoboards for first 100 leads.
    b. Receive response.
    c. Parse response with received recipients.
    d. Save recipients to Salesforce.
    e. Repeat till end of Recipient list.
    f. Send request to /marketing/searchleads for first 150 leads.

    g. Receive response.
    h. Parse response with received leads.
    i. Save leads to Salesforce.
    j. Repeat till end of Leads list.
4. Every X minutes the ViewsUpdateBatch send request to `/api/v2/views` for updating Views, Features and Question Answer data.
5. Every X minutes the DemoBoardCrmSyncBatch send request to `/api/integr/v1.0/link/updateLinkExternalOppAccData` for updating account/opportunity Ids in Consensus for merged accounts in Salesforce.

Schedulers features:

1. Sync DemoBoard data from - date picker input. If blank the schedulers will syn all the data during the daily sync. If a date selected the sync process will sync only the DemoBoard data created after the selected date. This features is designed for orgs with large amount of Consensus data needs to be synced;
2. Time interval for recent Data sync - integer input. Set how often the Recent data sync batches will be executing as 10 minutes is the minimum value and 240 is the maximum value for this input (the default value is 10);
3. Time interval for full Data sync - integer input. Set how often the Daily sync data batches will be executing as 1 is the minimum value and 7 is the maximum value for this fields (default is 1 - full sync data will run once per day).
4. Enable Task Creation - checkbox. If active a task records will be created, If inactive no tasks will be created in the org (active is the default value).

The RecentDemosUpdateBatch scheduled apex job will only start running as configured once a successful full sync of data has been processed by running the manual “Run Data Update” button on the configuration page.

Webhooks

Steps:

1. After event happened (We have 2 events: Demo watched (Recipient) and New lead (Lead))
2. Consensus side sends response with information.
3. Salesforce side parses this information.
4. Creates new lead.
5. Saves information to created lead**.**

For creating Lead via Webhook:

1. For demo_watched event the following fields are set on the Lead record:
    a. Email

b. FirstName

c. LastName

d. Title

e. Company

f. existInConsensus__c is set to *true*



2. For new_lead event the following fields are set on the Lead record:

a. Email

b. FirstName

c. LastName

d. Title

e. Company

f. existInConsensus__c is set to *true*

g. Country

h. State

i. Phone

j. Description



By Webhooks only Lead will be created, without additional information about Demo was watched.

Additional information about Demo name/Feature rating and others are requested a little bit later via Schedulers if they are running. (or via Run Data Update button)

Create DemoBoard from Consensus Component within Salesforce

After click on “Create DemoBoard” button from the Consensus component:

1. A modal opens the Consensus plugin through iframe, where the user can create Demoboard directly in Consensus;
2. All the DemoBoards are created back to Salesforce with the batch sync process (Run Data Update from Configuration page).

Related functionalities around “Create DemoBoard“:

1. When the user log in through the iframe in Consensus, the system creates a User Info table record;
2. If the creation of the DemoBoard is initiated from account or opportunity record, the user can select a recipients from the related contacts list;
3. When the iframe is loaded we’re passing some data with query parameters to be pre-populated in Consensus:
   a. source - the DemoBoard creation source (“sfdc“);
   b. accountId - if the DemoBoard creation is NOT initiated from a lead record;
   c. accountName - same as b.;
   d. opportunityId - if the DemoBoard creation is initiated from Opportunity record;
   e. opportunityName - same as d.;
   f. contacts - based on the target record it contains all related contacts for account or opportunity, or contact, or lead.
4. When the user click on “Create Demoboard“ in the iframe/plugin we receive an event with the organization value and all the recipients added to the demoboard in Consensus (this is in case the user change the pre-populated recipients);
5. When the user click on Insert Button/Link/Thumbnail in the iframe/plugin we receive an event with the Demoboard link and we redirect the user to Send Email global action with pre-populated email template within the DemoBoard link and the recipients;

CRM Sync

CRM Sync is the functionality to get accounts, opportunities and contacts during DemoBoard creation in Consensus. We are using a global/shared Connected App with OAuth settings to authorize the user and get the data from Salesforce. Only one admin user for both systems - Consensus and Salesforce, can connect/authorize and all other Consensus users can get the data on behalf of this user. Record visibility is applied so each user can see only the data it has access to.

CRM Sync flow:

1. Initiate the connection from Consensus - Integrations → Salesforce → Connect;
2. Type the Salesforce URL from which account and opportunities should be selected;
3. Allow the scopes of our Connected App;
4. If no errors the connection will be setup properly and an access token will be saved in Consensus;
5. All Consensus users in this environment should be able to select account and opportunity during DemoBoard creation.

NOTE: For a user with different email between Consensus and Salesforce, User Info records should be created to match the emails between the systems, otherwise selecting account and opportunity during the DemoBoards creation won't work.

Opportunity Recommendations

Opportunity Recommendations is a functionality to automatically select the best opportunity for the DemoBoards during creation in Consensus. The flow fires every time a recipient is added on the DemoBoard and tries to find the best opportunity based on several rules. If such opportunity is found it is automatically pre-populated on the DemoBoard.

Opportunity Recommendation rules:

Find all contacts by email in Salesforce and get all opportunities linked to this contacts, then the result list filter by account(if selected), by Is Primary contact, then by opportunity stage (probability), then by opportunity created date (desc). Pre-select the first one from the list.

Report Types

- Consensus Contacts (Accounts & Contacts category) - includes fields from objects: Contacts, Invitations, Views, Features;
- Consensus DemoBoards (Other Reports category) - includes fields from objects: DemoBoards, Invitations, Views, Features;
- Consensus Leads (Leads category) - includes fields from objects: Leads, Invitations, Views, Features;
- Consensus Opportunities (Opportunities category) - includes fields from objects: Opportunities, DemoBoards, Invitations, Views;
- Consensus Recipients (Lead category) - includes fields from Invitation object;
- Consensus Views (Other Reports category) - includes fields from View object;

Reports

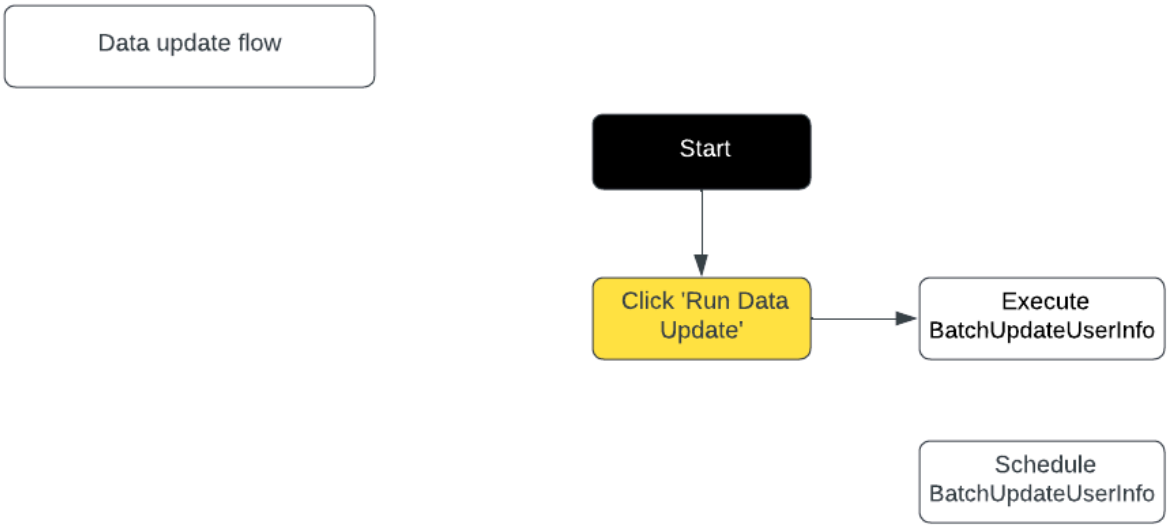
- Best Demos List - shows a list of demos with Total DemoBoards, Total Views and Total Shares columns;
- Consensus Demo Views - shows a list of views grouped by DemoBoards;
- Consensus Pipeline Influence - shows Sum Amount of all OPEN Opportunities where Consensus was used with at least 1 VIEWED DemoBoard;
- Consensus Recipients Demolytics - shows Demolytics data for the Consensus recipients grouped by Account;
- Consensus View Timeline - shows views data by created date;
- DemoBoards Sales Demolytics - shows DemoBoard Demolytics data grouped by with/without views for comparison
- Total DemoBoards Created - shows all DemoBoards grouped by Opportunity;
- Monthly Recipients Report - shows demos with invitations list;
- Opportunities w/ DemoBoards Report - shows opportunity with DemoBoards;
- Consensus Won ARR/ACV - shows Sum Amount of CLOSED WON Opportunities where Consensus was used with at least 1 VIEWED DemoBoard;
- Non-Consensus Won ARR/ACV - shows CLOSED WON Opportunities where Consensus did NOT have any influence. 0 views and 0 view time;
- Closed Opportunity Report- shows closed opportunities grouped by with/without views for comparison;
- User Sales Demolytics - shows Demolytics data by DemoBoard owner.

Dashboards

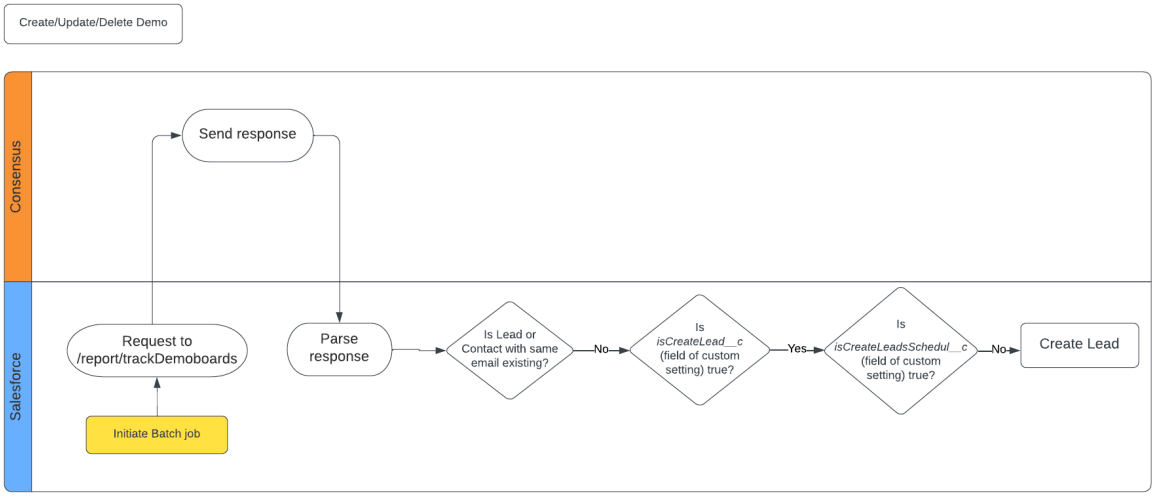
- Consensus ROI Dashboards
- Consensus Sales Demolytics

Data Flow Diagrams

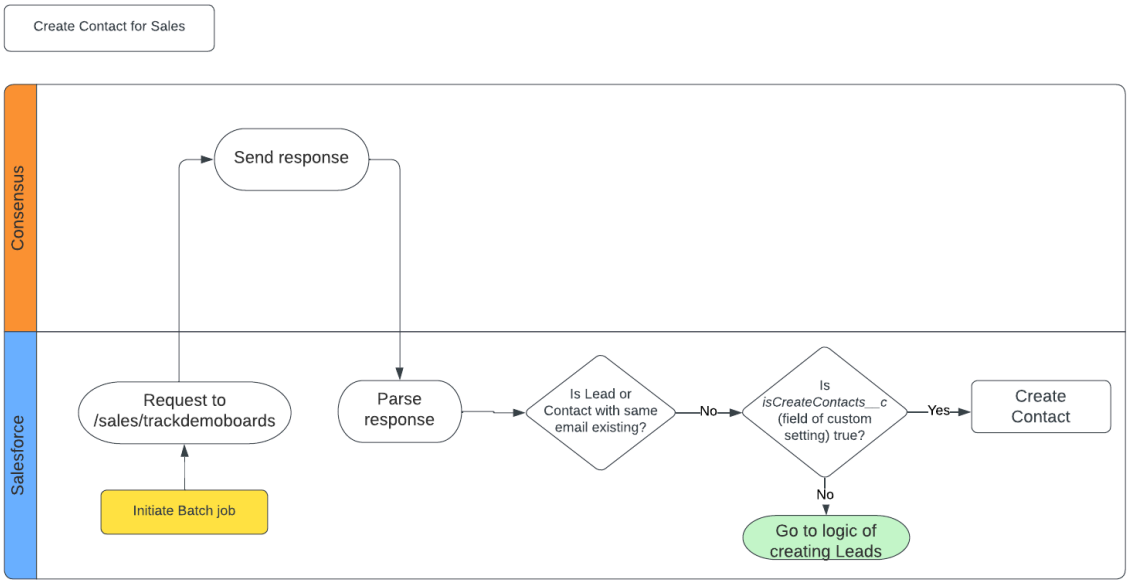
Run Date Update

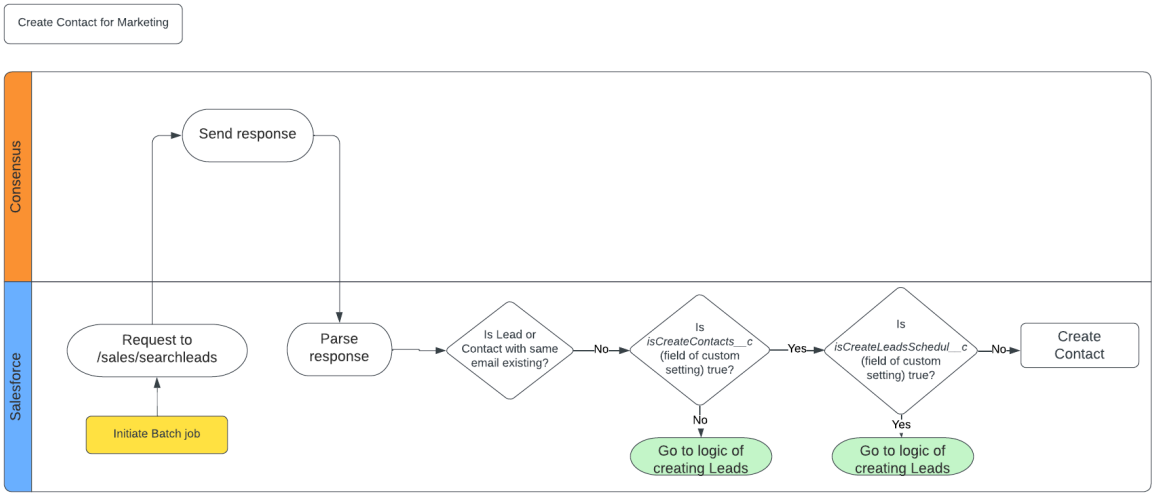


Delete Demos, which were deleted in Consensus.

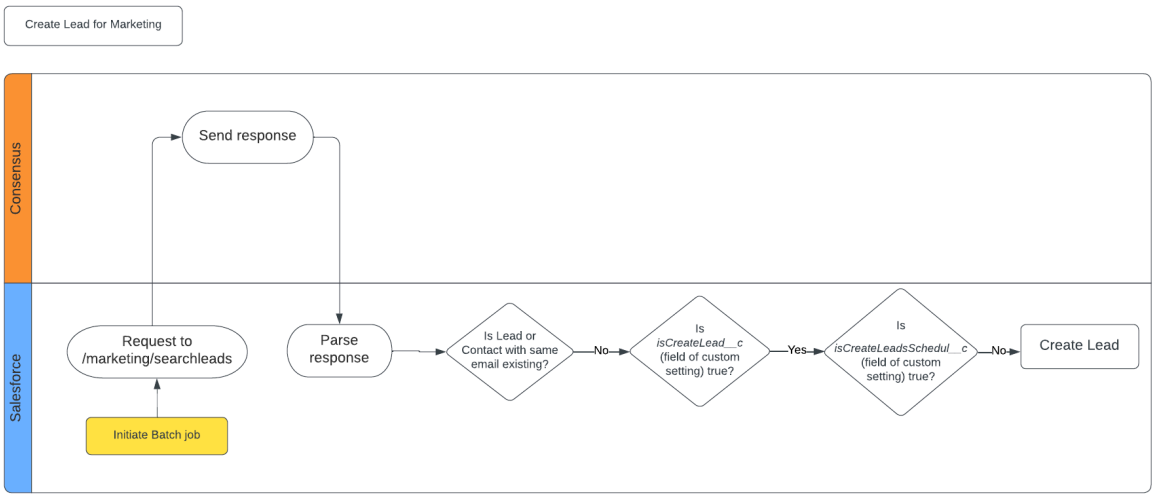


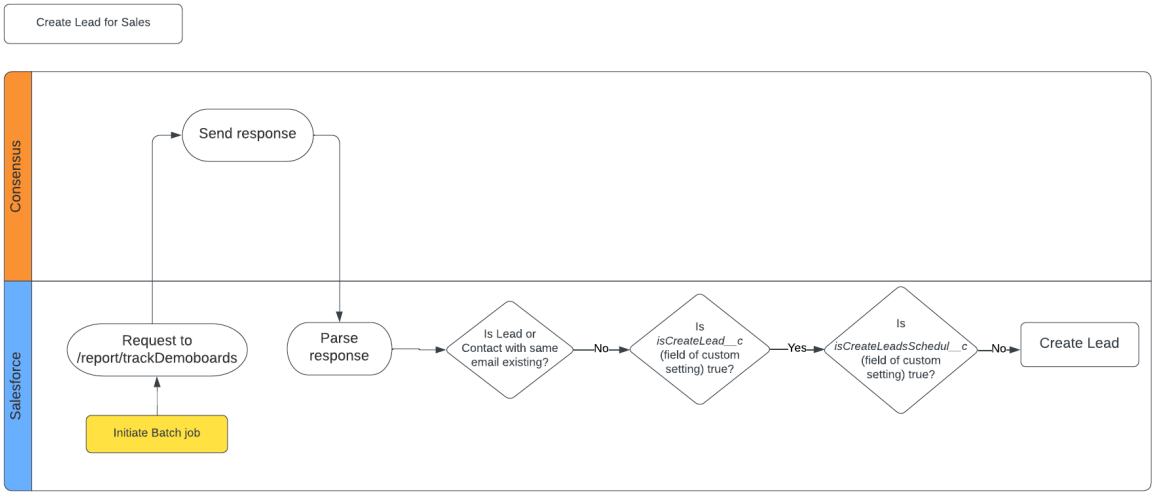
Contact Creation


Create Contact for Sales
Consensus
Salesforce
Send response
Request to /sales/trackdemoboards
Initiate Batch job
Parse response
Is Lead or Contact with same email existing?
No
Is isCreateContacts__c (field of custom setting) true?
Yes
Create Contact
No
Go to logic of creating Leads

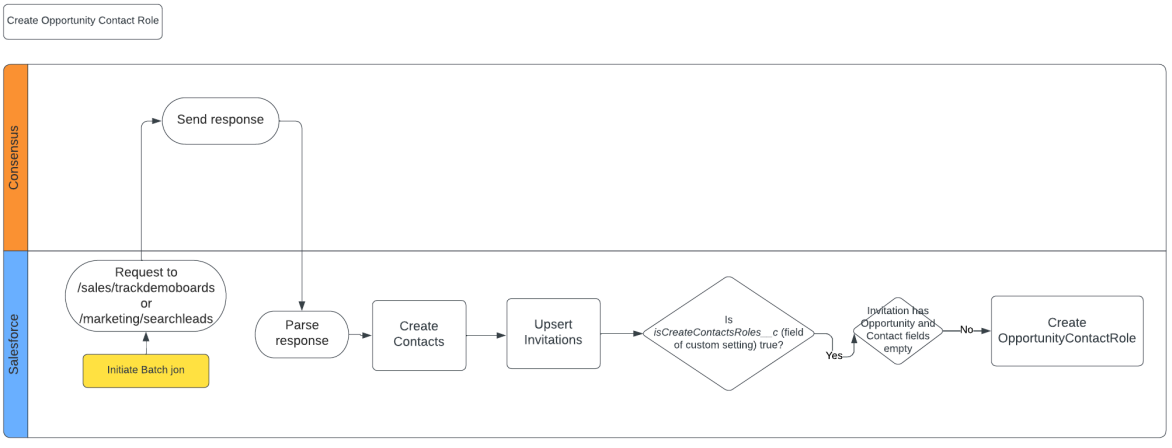

Create Contact for Marketing
Consensus
Salesforce
Send response
Request to /sales/searchleads
Initiate Batch job
Parse response
Is Lead or Contact with same email existing?
No
Is isCreateContacts__c (field of custom setting) true?
Yes
Is isCreateLeadsSchedul__c (field of custom setting) true?
No
Create Contact
No
Go to logic of creating Leads
Yes
Go to logic of creating Leads

Create Lead

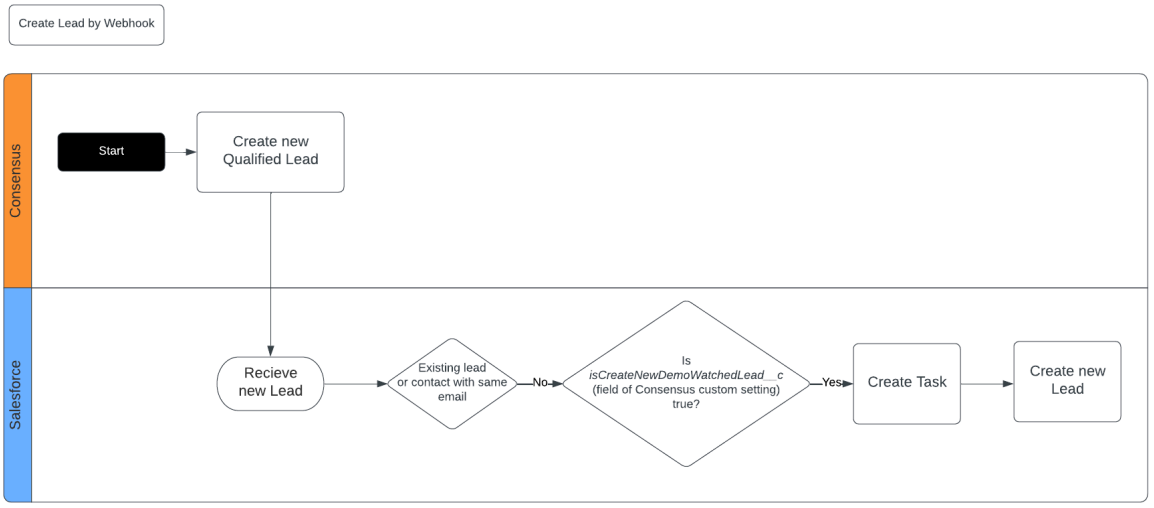




Contact Role Creation



Create Lead by Webhook



Saleo Integration for Consensus

Overview

The Saleo integration extends the Consensus Salesforce managed package to support revenue attribution for Saleo Live and Saleo Demo Agent sessions. The integration adds two boolean fields to the standard Account object. Saleo reads and writes data between its backend and the customer's Salesforce instance through the Consensus Connected App, using a separate OAuth consent and token for each Saleo customer. No additional custom objects, flows, triggers, webhooks, or batch jobs are introduced by this integration. Revenue attribution calculations run on Saleo's backend using data queried from Salesforce in real time.

Requires Consensus Salesforce managed package version 11.13 or later. The two Account fields ship in version 11.13; customers on earlier versions will have the integration detected as installed but flag writes will fail until the package is upgraded.

Data schema (with explanation of fields)

Standard Objects

Account

API Name	Type	Value
Consensus__Saleo_Influenced__c	Boolean	TRUE while at least one Saleo Live session is assigned to

		the Account
Consensus__Saleo_Agent_Influenced__c	Boolean	TRUE while at least one Saleo Demo Agent session is assigned to the Account

Custom Objects

None introduced by the Saleo integration.

Permission Sets

The Saleo integration reuses the existing Consensus Integration permission sets (Consensus Admin, Consensus Integration and Consensus Users PSG). The following permissions are added to that permission set:

- Field-level Edit on Consensus__Saleo_Influenced__c (Account)
- Field-level Edit on Consensus__Saleo_Agent_Influenced__c (Account)

Note: all other permissions required by the Saleo integration (Read on Account, Read on Opportunity, Read on User, API Enabled, ~~View Setup and Configuration~~) are already included in the existing Consensus permission sets.

Authentication

Saleo authenticates to the customer's Salesforce instance using the Consensus Connected App via standard OAuth 2.0. Each Saleo customer completes a separate OAuth consent flow and is issued a distinct OAuth token, scoped to that customer.

Required Scopes

- api
- refresh_token (offline access)

Callback URLs

- Production: `https://app.platform.saleo.io/oauth?type=salesforce`
- Staging: `https://app.staging-saleo.io/oauth?type=salesforce`

API Endpoints

All requests Saleo makes to Salesforce use the standard Salesforce REST API (v64.0). No custom Apex REST endpoints are required.

Authentication

- POST /services/oauth2/token (auth code + refresh)
- POST /services/oauth2/revoke
- GET identity URL

Organization

- GET /query?q=SELECT Name FROM Organization (executed at connect)

Object metadata

- GET /sobjects/Account/describe
- GET /sobjects/Opportunity/describe

User queries

- GET /query?q=SELECT Id, Name, Email FROM User WHERE IsActive = true

Account queries

- GET /query?q=SELECT Id, <name>, <domain>, <flag> FROM Account ORDER BY LastModifiedDate DESC (paginated)

Opportunity queries

- GET /query?q=SELECT Id, CreatedDate, IsWon, IsClosed, <closeDate>, <amount> FROM Opportunity WHERE AccountId = :id AND (IsWon = true OR IsClosed = false)
- GET /query?q=SELECT Id, IsWon, IsClosed, CloseDate, CreatedDate, AccountId FROM Opportunity WHERE IsClosed = true AND IsWon != null

Package detection

- GET /tooling/query?q=... FROM InstalledSubscriberPackage

Flag write

- PATCH /sobjects/Account/{id} with body `{ "<flag>": true|false }`

Data Flow

Flag write on session completion

1. A Saleo session (Live or Demo Agent) completes on Saleo's backend.
2. Preview sessions are skipped. All other sessions proceed.
3. Saleo matches the session to a Salesforce Account using source-specific logic:
   - Live sessions: The demo presenter explicitly selects the Account for the session, either at the time of the demo or retroactively via Saleo's revenue analytics page.
   - Demo Agent sessions: Saleo attempts to auto-match by comparing the welcome-form email domain against the Account's Website domain. If no match is found, the session lands unassigned and may be manually mapped later via Saleo's revenue analytics page.
4. Saleo issues a PATCH request to /sobjects/Account/{id}, setting Consensus__Saleo_Influenced__c = TRUE for Live sessions, or Consensus__Saleo_Agent_Influenced__c = TRUE for Demo Agent sessions.

5. Per-account session counts are maintained in Saleo's backend. If all sessions associated with an Account are later unmapped, Saleo issues a PATCH request to set the corresponding flag back to FALSE.

Revenue analytics calculation

1. A user opens the Saleo Revenue Analytics dashboard.
2. Saleo's backend queries the customer's Salesforce instance via the REST API.
3. Saleo takes the set of Accounts that have at least one assigned session (from its own backend, not from the Salesforce flag) and, for each, queries that Account's Opportunities that are Closed Won or still open.
4. Attribution logic runs on Saleo's backend. An Opportunity is considered influenced if it was open at the time of the session or created within 30 days after the session, and the Opportunity Close Date is on or after the session date. Duplicate matches are counted once.
5. Results are rendered in the Saleo dashboard. Salesforce data read during this process is not persisted to Saleo's database.

Note: attribution logic is identical for Live and Demo Agent sessions. Per-source counts are maintained independently so Live and Demo Agent influence are tracked and reported separately.

Security Considerations

Data location

Session data is stored in Saleo's backend infrastructure. Salesforce data read by Saleo during revenue analytics calculation is not persisted; results are computed on demand and rendered to the user.

The two Account boolean fields written by Saleo are the only data Saleo writes to the customer's Salesforce instance. No Lead, Contact, Opportunity, Task, or custom object records are created or modified by this integration.

Encryption

- In transit: all communication with Salesforce uses HTTPS (TLS 1.2+).
- At rest: Saleo's backend data is stored in DynamoDB, which is encrypted at rest.
- OAuth tokens are encrypted using AWS KMS and scoped per customer.

Data retention

- OAuth tokens persist for the duration of the active integration. On disconnect, tokens are revoked via POST /services/oauth2/revoke and cleared from Saleo's backend.
- Account flags written by Saleo remain in the customer's Salesforce instance after disconnect and are not cleared automatically. The customer may remove them manually

via Salesforce administration.

- Session data in Saleo's backend follows Saleo's standard data retention policy. See Saleo Data Processing Agreement for details.

Access control

- All API access to the customer's Salesforce instance is authenticated via OAuth 2.0 and scoped by the permissions assigned to the Integration permission set.
- Saleo cannot access any Salesforce data outside the scopes granted during OAuth consent.
- The Integration user may be assigned the Salesforce Integration license (API Only, Minimum Access profile) to restrict login access to API activity only.