



School Leadership Focus Reporting

Technical Reference

Version 1.0 | April 2026

1. System Inventory

The following tables enumerate every component of the School Leadership Focus Reporting system. All components reside within the Clear River Public Schools Microsoft 365 tenant.

1.1 Microsoft 365 Environment

Item	Value	Notes
Tenant Domain	crps.co.us	Primary M365 tenant for the portfolio demonstration environment
M365 Subscription	Microsoft 365 Business Standard	Includes Power Apps, Power Automate (standard connectors), SharePoint, Teams, and Exchange
Fabric Subscription	Microsoft Fabric (free)	Required for all users
Power BI Subscription	Power BI Pro	Required for all users. The solution only uses standard features (no premium feature dependencies).
Power BI Workspace	SL_FocusReporting	Hosts both the published semantic model and the thin report
Power Apps Subscription	Microsoft Power Apps for Developer	Required for all users
Power Automate Subscription	Microsoft Power Automate Free	Required for users maintaining the Power Automate flow
SharePoint Site	https://crps.sharepoint.com/sites/SL_FocusReporting	Hosts the Submissions and ReportingPeriods lists
Shared Mailbox	school_leadership@crps.co.us	Used as the sender for all automated reminder emails. No user license required.
On-Premises Gateway	Installed on shared server	Required for Db2 data refresh. Must be running for scheduled or manual refresh to succeed.

1.2 Entra ID Security Groups

Group	Role	Purpose
sl_sec_teammember@crps.co.us	Team Member	Members receive the TeamMember Power BI RLS role, which restricts data access to their own records only. Drives the Power Automate reminder flow's team member list.
sl_sec_management@crps.co.us	Management	Members receive no Power BI RLS restriction (full data access). Used by the reminder flow to identify CC recipients and to exclude managers from the compliance check.

1.3 SharePoint Lists

1.3.1 Submissions

Stores weekly time log entries submitted by team members via the Power App.

Column	Type	Description
Title	Single line of text	Auto-generated key: '{email} - {WeekEndingDate}'
SubmittedByEmail	Single line of text	UPN of the submitting user. Used for RLS filtering and Power Automate anti-join.
WeekEndingDate	Date	Saturday date of the reporting week. Stored as date only (no time component).
SchoolYear	Single line of text	School year label (e.g. '2025-2026'). Populated by the Power App at submission time.
hrs_InstructionalLeadership	Number	Hours logged for Instructional Leadership category.
hrs_Operations	Number	Hours logged for Operations category.
hrs_ProfDev	Number	Hours logged for Professional Development category.
hrs_CommunityEngagement	Number	Hours logged for Community Engagement category.
hrs_Other	Number	Hours logged for Other category.
hrs_Total	Number	Sum of all five category hours.
SubmissionStatus	Choice	Values: Submitted, Draft. Currently all submissions are Submitted; Draft is reserved for a future phase.
Notes	Multiple lines	Optional free-text notes from the team member.

Item-level permissions: *Read* access is set to **Items created by the user**. Each team member can only read their own submissions. This is critical — seeded or bulk-loaded data must be loaded under the intended user's credentials or it will be invisible to that user.

1.3.2 ReportingPeriods

Reference list of all valid weekly reporting periods (Saturdays) for the current and prior school years. Used by the Power App to build the week selector dropdown and by the Power Automate flows for compliance checking.

Column	Type	Description
Title	Single line of text	Human-readable label (e.g. 'Week ending Sep 06, 2025').
PeriodDate	Date	Saturday date of the reporting period. Stored with 1:00 AM time component by SharePoint — this is handled in the Power Automate flow's date comparison logic.
SchoolYear	Single line of text	School year label (e.g. '2025-2026').
SortOrder	Number	Integer sort key for ordered display in the Power App dropdown.
IsActive	Yes/No	Flag to enable/disable a period without deleting it. Reserved for future use.

Annual maintenance: At the start of each school year, add the new year's Saturday dates to this list. This is the only required annual maintenance task for the system.

1.4 Power BI Semantic Model

1.4.1 Tables

Table	Source	Description
Submissions	SharePoint (Import)	Weekly time log entries. One row per team member per week.
TA_LeaveData	IBM Db2 via on-premises gateway (Import)	Weekly leave hours per employee. One row per employee per week. Filtered to school year start date or later.
DateTable	DAX calculated table	Standard date dimension spanning September 2024 through December 2028 (dynamic end date: YEAR(TODAY())+2). One row per calendar day.
WeekBridge	DAX calculated table	One row per Saturday (week-ending date). Acts as the hub table connecting Submissions and TA_LeaveData. Contains school year, week number, and flag columns.
Users	Power Query (SharePoint connector)	Distinct list of email addresses derived from the union of Submissions[SubmittedByEmail] and TA_LeaveData[EMPLOYEE_EMAIL]. Auto-refreshes with the model. Used as the person dimension for RLS.
_Measures	DAX (no data rows)	Placeholder table housing all measures. Contains a single hidden dummy column.

1.4.2 Relationships

From (Many)		To (One)	Direction	Active
Submissions[WeekEndingDate]	→	WeekBridge[WeekEndingDate]	Single	Yes
TA_LeaveData[WEEK_ENDING_DATE]	→	WeekBridge[WeekEndingDate]	Single	Yes
DateTable[Week Ending Date]	→	WeekBridge[WeekEndingDate]	Single	Yes
Submissions[SubmittedByEmail]	→	Users[Email]	Single	Yes
TA_LeaveData[EMPLOYEE_EMAIL]	→	Users[Email]	Single	Yes

WeekBridge is the hub table. All filters from **DateTable** school year columns reach **TA_LeaveData** and **Submissions** only via **WeekBridge** relationships.

1.4.3 Row-Level Security (RLS)

Role	Filter Expression	Effect
TeamMember	Users[Email] = USERPRINCIPALNAME()	Restricts the Users table to the signed-in user's row. Filter propagates via relationships to Submissions and TA_LeaveData, isolating all data to the current user only.
Management	(none)	No filter applied. Managers see all data for all team members.

RLS is applied at the **Users** table (dimension/one-side), not the fact tables directly. REMOVEFILTERS(Users[Email]) in team average measures cannot override RLS — this is a Power BI security design constraint, not a bug.

1.5 Power App

Component	Type	Description
School Leadership Focus Reporting	Canvas App	Main application. Two screens accessible to all users; report content is access-controlled by Power BI RLS.
scrSubmit	Screen	Weekly submission form. Displays a week selector dropdown (missing weeks only) and hour entry fields for each category. Submission writes to the Submissions SharePoint list.
scrReports	Screen	Embeds the Power BI thin report via the Power BI Tile control using TileUrl with AllowNewAPI = true.
cmpNavBar	Component	Reusable navigation bar shown at the top of all screens. Navigation buttons route between scrSubmit and scrReports.

1.6 Power Automate Flows

Flow	Trigger	Description
LTT - Monday Reminder	Weekly, Monday 12:00 PM MT	Scheduled flow that executes the full submission compliance check and sends reminder emails to non-compliant team members.
LTT - Tuesday-Friday Reminder	Weekly, Tue–Fri 8:00 AM MT	Identical logic to the Monday flow. Runs daily at 8 AM Tuesday through Friday.

Both flows contain identical logic. A child flow architecture was designed but not implemented due to a legacy designer compatibility constraint in the current Power Automate environment. See [Section 7 \(Decision Log\)](#) for details.

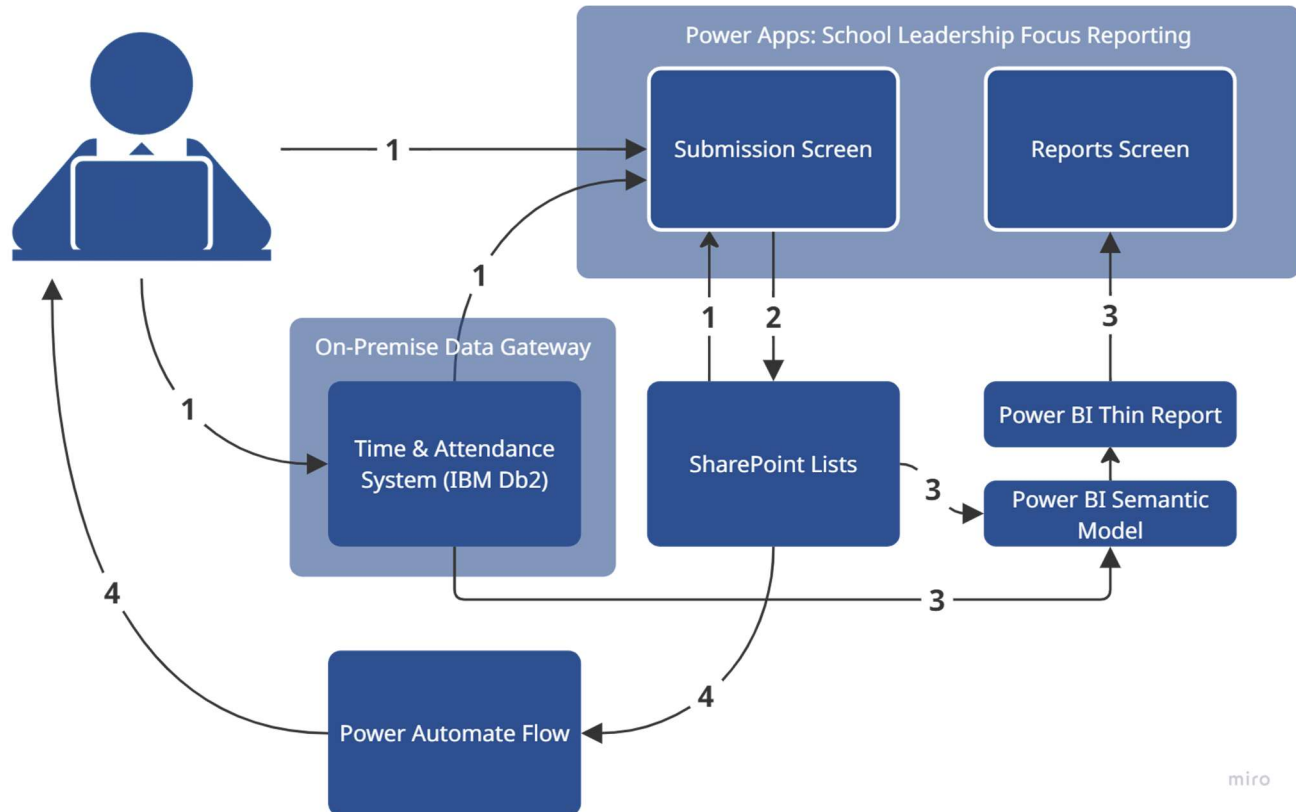
2. Data Architecture

2.1 Source Systems

Source	System	Connection Method	Data Supplied
Submission data	SharePoint Online	SharePoint connector (cloud, no gateway)	Weekly time log entries from team members
Reporting Period data	SharePoint Online for Power Apps & Power Automate	SharePoint connector (cloud, no gateway)	Reporting period dates and associated metadata (e.g. the school year that the period falls in)
	DAX calculated table for Power BI	N/A – internal calculation	
Leave data	IBM Db2	On-premises data gateway (required)	Holiday, PTO, sick, and total leave hours per employee per week
User profiles	Microsoft 365 / Power Apps	Office365Users connector	User profile data
Security groups	Microsoft Entra ID	Office 365 Groups connector	Team member and management group membership for reminder targeting

2.2 Data Flow

The diagram below shows the flow of data through all system components.



Data flow narrative:

1. The Submissions screen of the app pulls data from the SharePoint lists and displays a list of weeks that don't yet have submission data. The app also pulls leave data from the time and attendance system for the selected week and displays it to the team member for informational purposes (team members maintain leave data in the attendance system as a separate process). The team member logs hours for the week in the relevant categories and submits the data.
2. The app writes the data to the SharePoint **Submissions** list.
3. The Power BI semantic model is set to refresh daily. The report is embedded in the Reports screen of the app.
4. Power Automate reads the submission data from the SharePoint list once a day (weekdays) and sends reminder emails to team members for missing submissions.

2.3 Refresh Architecture

Component	Refresh Type	Notes
Submissions (SharePoint)	Scheduled and manual	The model is scheduled for daily (weekdays) refresh. Manual refreshed can be triggered in Power BI Service up to 7 times per day (8 total per day) with current licensing level (Power BI Pro). Note that both connections must be routed through the On-Premises Gateway in order to schedule refreshed in Power BI service.
TA_LeaveData (Db2)		
DateTable	On model refresh	DAX calculated table. Recalculates automatically on every model refresh. End date is dynamic: YEAR(TODAY())+2.
WeekBridge	On model refresh	DAX calculated table. Recalculates on every model refresh. Note that columns IsFutureWeek, IsCurrentSchoolYear, and IsCurrentReportingWeek use TODAY() and are therefore refresh-time snapshots - they do not auto-update between refreshes.
Users	On model refresh	Power Query table derived from Submissions and TA_LeaveData. Refreshes automatically when the model refreshes.

Refresh Failure Impact: Calculated Table Staleness

WeekBridge columns that use TODAY() (IsFutureWeek, IsCurrentSchoolYear, IsCurrentReportingWeek) are computed at refresh time, not at query time. If the model is not refreshed daily (e.g. if a refresh fails and is not re-triggered), these flags may be one or more days stale.

Impact: IsFutureWeek may incorrectly include or exclude the current week on the day it transitions.

Workaround in use: All cumulative measures compute their own cutoff logic using TODAY() at query time, independent of WeekBridge columns. Only visual-level filters on IsFutureWeek are affected by staleness.

3. Security Model

3.1 Access Control Overview

Access to the system is controlled at three levels:

- Microsoft 365 security group membership (determines RLS role)
- Power BI workspace permissions (determines who can view/edit the report)
- SharePoint list permissions (determines who can read/write submission data)

User	Security Group	Power BI Role	Data Visible
Team members	sl_sec_teammember	TeamMember	Own submissions and leave data only. Cannot see other team members' data in any visual.
Manager	sl_sec_management	Management (no RLS)	All team members' data. Full report access including team averages and compliance across all members.
Executive assistants	sl_sec_management	Management (no RLS)	Same as manager. Receives Cc on all reminder emails.

3.2 RLS Design Details

The system uses Option A RLS architecture — the security filter is applied to the **Users** dimension table rather than directly to the fact tables. This approach was chosen over Option B (direct fact table filters) because it allows relationship propagation to handle data isolation cleanly, and because the **Users** table was already needed as a person dimension for report slicers.

Filter propagation chain under **TeamMember** RLS:

- Users[Email] = USERPRINCIPALNAME() → filters Users to one row
 - Users[Email] → Submissions[SubmittedByEmail] (via relationship) → isolates submissions to current user
 - Users[Email] → TA_LeaveData[EMPLOYEE_EMAIL] (via relationship) → isolates leave data to current user

Important constraints:

- **REMOVEFILTERS(Users[Email])** in DAX measures does not override RLS. This is a Power BI security feature - RLS filters cannot be removed by measure-level DAX. The **ILRP - Team Avg** and **ILFT - Team Avg** measures therefore do not function correctly for **TeamMember** role users. These measures have been set as not-visible in the model and team average calculations are achieved by slicing the reports by multiple users in *Management View* mode.
- The Power BI Desktop *View As Role* feature does not reliably test composite model RLS. Always test RLS by publishing to the Service and viewing as the target user.

3.3 SharePoint Permissions

List	Setting	Effect
Submissions	Item-level read: items created by user	Each team member can only read their own submission records. Critical: bulk-loaded data must be inserted under the intended user's credentials.
Submissions	Item-level create: all items	Team members can create new submission records through the Power App.
ReportingPeriods	Read: all users	All users can read the reporting periods list — required for the Power App week selector dropdown.
ReportingPeriods	Write: site owners only	Only site admins can add or modify reporting period entries.

4. Measure Catalog

The following table documents all production measures in the semantic model. Hidden supporting measures are listed at the end. Test measures (display folder: *Test Measures*) are excluded from this catalog.

All measures live on the **_Measures** table. Display folders organize measures within the field pane but do not affect calculation or performance.

4.1 Global Variables

Current School Year	Returns the school year label (e.g. '2025-2026') for today's date, ignoring all filter context.
	<pre>CALCULATE(LOOKUPVALUE(DataTable[School Year], DataTable[Date], TODAY()), ALL())</pre>
	Dependencies: DataTable Note: Uses ALL() to ensure this is never affected by report slicers or filters.
Current Reporting Week	Returns the week-ending date (Saturday) of the most recent elapsed reporting week.
	<pre>VAR varSat = TODAY() + 7 - WEEKDAY(TODAY(), 1) VAR varPrevSat = TODAY() - WEEKDAY(TODAY(), 1) RETURN IF(TODAY() = varSat, varSat, varPrevSat)</pre>
	Dependencies: None Note: If today is Saturday, returns today. Otherwise returns the prior Saturday.
Current Reporting Week Label	Returns the current reporting week as an uppercase formatted date string (e.g. 'SAT 4/19/2026') for display in report visuals.
	<pre>UPPER(FORMAT(CALCULATE(LOOKUPVALUE(WeekBridge[WeekEndingDate], WeekBridge[IsCurrentReportingWeek], TRUE()), ALL(WeekBridge)), "DDD M/D/YYYY"))</pre>
	Dependencies: Current Reporting Week (measure), WeekBridge (table)

4.2 Worked Hour Totals

WH - IL	Sum of Instructional Leadership hours in the current filter context.
	SUM(Submissions[hrs_InstructionalLeadership])
	Dependencies: Submissions
WH - Ops	Sum of Operations hours.
	SUM(Submissions[hrs_Operations])
	Dependencies: Submissions
WH - PD	Sum of Professional Development hours.
	SUM(Submissions[hrs_ProfDev])
	Dependencies: Submissions
WH - CE	Sum of Community Engagement hours.
	SUM(Submissions[hrs_CommunityEngagement])
	Dependencies: Submissions
WH - Oth	Sum of Other hours.
	SUM(Submissions[hrs_Other])
	Dependencies: Submissions
WH - Total	Sum of all submitted hours.
	SUM(Submissions[hrs_Total])
	Dependencies: Submissions
WH - Non-IL	Total non-Instructional Leadership hours (all categories except IL).
	[WH - Total] - [WH - IL]
	Dependencies: [WH - Total], [WH - IL]

4.3 Leave Hours

Leave Hrs - Total	Sum of all leave hours (holiday + PTO + sick).
	SUM(TA_LeaveData[TOTAL_LEAVE_HOURS])
	Dependencies: TA_LeaveData
Leave Hrs - Holiday	Sum of holiday leave hours.
	SUM(TA_LeaveData[HOLIDAY_HOURS])
	Dependencies: TA_LeaveData
Leave Hrs - PTO	Sum of PTO leave hours.
	SUM(TA_LeaveData[PTO_HOURS])
	Dependencies: TA_LeaveData
Leave Hrs - Sick	Sum of sick leave hours.
	SUM(TA_LeaveData[SICK_HOURS])
	Dependencies: TA_LeaveData
Leave Hrs - Non-Holiday	Sum of PTO and sick leave (excludes holidays).
	[Leave Hrs - PTO] + [Leave Hrs - Sick]
	Dependencies: [Leave Hrs - PTO], [Leave Hrs - Sick]

4.4 Worked Hour Percentages

% IL	IL hours as a percentage of total hours. Alias for ILRP, used for the percentage stacked column visual.
	<code>DIVIDE([WH - IL], [WH - Total], BLANK())</code>
	Dependencies: [WH - IL], [WH - Total] Note: Returns <code>BLANK()</code> (not 0) when <code>WH - Total = 0</code> so zero-hour weeks are excluded from averages.
% Non-IL	Non-IL hours as a percentage of total hours.
	<code>IF([WH - Total] = 0, BLANK(), 1 - [% IL])</code>
	Dependencies: [WH - Total], [% IL] Note: Returns <code>BLANK()</code> for zero-hour weeks to prevent misleading 100% Non-IL bars on leave weeks.
% Operations	Operations hours as a percentage of total hours.
	<code>DIVIDE([WH - Ops], [WH - Total], 0)</code>
	Dependencies: [WH - Ops], [WH - Total]
% ProfDev	Professional Development hours as a percentage of total hours.
	<code>DIVIDE([WH - PD], [WH - Total], 0)</code>
	Dependencies: [WH - PD], [WH - Total]
% Community Engagement	Community Engagement hours as a percentage of total hours.
	<code>DIVIDE([WH - CE], [WH - Total], 0)</code>
	Dependencies: [WH - CE], [WH - Total]
% Other	Other hours as a percentage of total hours.
	<code>DIVIDE([WH - Oth], [WH - Total], 0)</code>
	Dependencies: [WH - Oth], [WH - Total]

4.5 KPI

ILRP	Instructional Leadership Relative Percentage. IL hours divided by total submitted hours. Measures what fraction of actual working time went to IL.
	<code>DIVIDE([WH - IL], [WH - Total], BLANK())</code>
	Dependencies: [WH - IL], [WH - Total] Note: BLANK() alternate result (not 0) is intentional — ensures zero-hour weeks are excluded from AVERAGEX calculations.
ILFT	Instructional Leadership Fixed Target. IL hours divided by a fixed baseline of minimum required hours (40 minus total leave). Accounts for leave taken.
	<code>DIVIDE([WH - IL], SUMX(TA_LeaveData, MAX(0, 40 - TA_LeaveData[TOTAL_LEAVE_HOURS])), BLANK())</code>
	Dependencies: [WH - IL], TA_LeaveData Note: SUMX iterates TA_LeaveData row by row, computing per-person available hours before summing. This correctly handles multi-person filter contexts in the Management view.
ILRP - Cumulative	Year-to-date cumulative ILRP. Accumulates IL hours and total hours from week 1 of the current school year through the current row's week.
	<code>VAR varYear = MAX(WeekBridge[SchoolYear]) VAR varWeekNo = MAX(WeekBridge[SchoolWeekNumber]) VAR varEnd = MAX(WeekBridge[WeekEndingDate]) VAR varFuture = varEnd > TODAY() && varYear = [Current School Year] VAR varWindow = FILTER(ALL(WeekBridge), WeekBridge[SchoolYear] = varYear && WeekBridge[SchoolWeekNumber] <= varWeekNo) VAR varIL = CALCULATE([WH - IL], varWindow) VAR varTotal = CALCULATE([WH - Total], varWindow) RETURN IF(varFuture, BLANK(), DIVIDE(varIL, varTotal, BLANK()))</code>
	Dependencies: [WH - IL], [WH - Total], [Current School Year], WeekBridge Note: Future weeks of the current school year return BLANK() so the YTD line terminates at the current week. Prior school year lines always render fully.
ILFT - Cumulative	Year-to-date cumulative ILFT. Accumulates IL hours against a fixed baseline of minimum required hours for all elapsed weeks.
	<code>VAR varYear = MAX(WeekBridge[SchoolYear]) VAR varWeekNo = MAX(WeekBridge[SchoolWeekNumber]) VAR varEnd = MAX(WeekBridge[WeekEndingDate]) VAR varFuture = varEnd > TODAY() && varYear = [Current School Year] VAR varWindow = FILTER(ALL(WeekBridge), WeekBridge[SchoolYear] = varYear && WeekBridge[SchoolWeekNumber] <= varWeekNo) VAR varIL = CALCULATE([WH - IL], varWindow) VAR varBaseline = CALCULATE(SUMX(TA_LeaveData, [ILFT - Cumulative Baseline]), varWindow) RETURN IF(varFuture, BLANK(), DIVIDE(varIL, varBaseline, BLANK()))</code>
	Dependencies: [WH - IL], [ILFT - Cumulative Baseline], [Current School Year], WeekBridge, TA_LeaveData

	Note: Uses ILFT - Cumulative Baseline as a supporting measure to keep the denominator logic readable. ALL(WeekBridge) is required to build the cumulative window independently of visual filter context.
ILRP - Team Avg	Average ILRP across all team members. Management report use only.
	<pre> CALCULATE(AVERAGEX(FILTER(VALUES(Submissions[SubmittedByEmail]), [WH - Total] > 0), [ILRP]), REMOVEFILTERS(Users[Email])) </pre>
	Dependencies: [ILRP], [WH - Total], Submissions, Users Note: REMOVEFILTERS(Users[Email]) lifts the person slicer filter in the Management view. Does not override RLS — this measure will only see each user's own data under TeamMember RLS.
ILFT - Team Avg	Average ILFT across all team members. Management report use only.
	<pre> CALCULATE(AVERAGEX(FILTER(VALUES(Submissions[SubmittedByEmail]), [WH - Total] > 0), [ILFT]), REMOVEFILTERS(Users[Email])) </pre>
	Dependencies: [ILFT], [WH - Total], Submissions, Users Note: See [ILRP - Team Avg] notes.
ILRP - Narrative	Text definition of ILRP for use as a tooltip field on report visuals.
	"Instructional Leadership Relative Percentage is the number of Instructional Leadership hours worked as a percentage of your total hours worked in the specified period (excludes all leave taken)."
	Dependencies: None
ILFT - Narrative	Text definition of ILFT for use as a tooltip field on report visuals.
	"Instructional Leadership Fixed Target is the number of Instructional Leadership hours as a percentage of a fixed target of total hours worked in a given period. The fixed target value for each week is calculated by subtracting all leave taken from 40."
	Dependencies: None
ILRP - Target Range	Static text '50% - 60%' for use as a tooltip field.
	"50% - 60%"
	Dependencies: None
ILFT - Target Range	Static text '50% - 60%' for use as a tooltip field.
	"50% - 60%"
	Dependencies: None

4.6 Support Measures

ILFT - Cumulative Baseline <i>Hidden</i>	Supporting measure only. Returns MAX(0, 40 - total leave hours) for the current TA_LeaveData row context. Used exclusively as the SUMX expression inside ILFT - Cumulative.
	MAX(0, 40 - SUM(TA_LeaveData[TOTAL_LEAVE_HOURS]))
	Dependencies: TA_LeaveData Note: Hidden. Do not use directly on report visuals. Designed to be called row-by-row inside a SUMX over TA_LeaveData.

4.7 Submission Compliance

Expected Weeks	Count of elapsed submission weeks in the current school year, summed across all users in the current filter context.
	<pre>VAR varYear = [Current School Year] RETURN SUMX(VALUES(Users[Email]), CALCULATE(COUNTROWS(WeekBridge), WeekBridge[SchoolYear] = varYear, WeekBridge[IsFutureWeek] = FALSE()))</pre>
	Dependencies: [Current School Year], WeekBridge, Users Note: Iterates over Users[Email] to correctly handle multi-person selections in the Management view.
Submitted Weeks	Count of weeks with at least one submission in the current school year, summed per user across all users in the current filter context.
	<pre>VAR varYear = [Current School Year] RETURN SUMX(VALUES(Users[Email]), CALCULATE(DISTINCTCOUNT(Submissions[WeekEndingDate]), WeekBridge[SchoolYear] = varYear, WeekBridge[IsFutureWeek] = FALSE()))</pre>
	Dependencies: [Current School Year], WeekBridge, Submissions, Users Note: DISTINCTCOUNT prevents duplicate submissions inflating the count.
Missing Weeks	Number of expected submission weeks with no submission.
	[Expected Weeks] - [Submitted Weeks]
	Dependencies: [Expected Weeks], [Submitted Weeks]
Compliance Rate	Submitted weeks as a percentage of expected weeks.
	DIVIDE([Submitted Weeks], [Expected Weeks], BLANK())
	Dependencies: [Submitted Weeks], [Expected Weeks] Note: Returns BLANK() (not 0) when Expected Weeks = 0, e.g. first day of the school year.

Submission Status	Returns '✓ Submitted' or 'X Missing' for a given week row. Used in the Compliance page missing weeks table.
	<pre> VAR varYear = [Current School Year] VAR varCutoff = TODAY() - 1 VAR varWeek = MAX(WeekBridge[WeekEndingDate]) VAR varFuture = varWeek > varCutoff VAR varScope = CALCULATE(COUNTROWS(DateTable), DateTable[Week Ending Date] = varWeek, DateTable[School Year] = varYear) = 0 VAR varHasSub = CALCULATE(COUNTROWS(Submissions), Submissions[WeekEndingDate] = varWeek) > 0 RETURN IF(varFuture varScope, BLANK(), IF(varHasSub, "✓ Submitted", "X Missing")) </pre>
	Dependencies: [Current School Year], WeekBridge, DateTable, Submissions Note: Uses TODAY() -1 cutoff so the current in-progress week is included once Saturday arrives.

5. Power Automate Flow Reference

Both reminder flows contain identical logic. The following sections document the shared logic and connector dependencies.

5.1 Flow Logic Summary

#	Step	Detail
1	Get Team Members	Calls Office 365 Groups connector to list members of <code>sl_sec_teammember</code> security group. Returns full user objects including <code>userPrincipalName</code> and <code>mail</code> .
2	Get Management Members	Calls Office 365 Groups connector to list members of <code>sl_sec_management</code> . Used for two purposes: building the Cc list and filtering management members out of the compliance check.
3	Get Management UPNs	Data Operations → Select. Extracts a flat array of UPN strings from the management members list. Used as the exclusion list in step 4.
4	Filter to Team Members Only	Data Operations → Filter Array. Removes any member whose UPN appears in the management UPN array. Result: only non-management team members proceed to the compliance check.
5	Get Management Emails	Data Operations → Select. Extracts email addresses from management members for use as Cc recipients.
6	Build CC String	Data Operations → Join. Joins management emails with semicolon separator, producing the Cc field value for Outlook.
7	Get Current School Year	Data Operations → Compose. Derives the current school year label using <code>formatDateTime</code> and string concatenation. Logic: if current month $\geq$ 09, school year = YYYY-(YYYY+1), else (YYYY-1)-YYYY.
8	Get Current Reporting Week Cutoff	Data Operations → Compose. Calculates midnight of the day after the most recent elapsed Saturday. Used as the <code>le</code> filter value in step 9. Formula: <code>addDays(startOfDay(addDays(utcNow(), mul(mod(add(dayOfWeek(utcNow()), 1), 7), -1))), 1)</code> .
9	Get Expected Periods YTD	SharePoint → Get Items (ReportingPeriods). Filter: <code>SchoolYear eq '[current year]'</code> and <code>PeriodDate le '[cutoff]'</code> . Top: 100.
10	Get Submissions YTD	SharePoint → Get Items (Submissions). Filter: <code>SchoolYear eq '[current year]'</code> . Top: 500.
11	For Each Team Member	Control → Apply to Each. Iterates over the filtered team members array from step 4.
12	Filter Submissions for Member	Inside loop. Data Operations → Filter Array. Isolates submissions for the current team member by comparing <code>SubmittedByEmail</code> to <code>userPrincipalName</code> .
13	Get Member Submitted Dates	Inside loop. Data Operations → Select. Extracts <code>weekEndingDate</code> values from the member's submissions, formatted as <code>yyyy-MM-dd</code> for string comparison.

#	Step	Detail
14	Find Missing Periods	Inside loop. Data Operations → Filter Array. Anti-join: keeps only expected periods where the formatted <code>PeriodDate</code> does not appear in the submitted dates array.
15	Format Missing Period Dates	Inside loop. Data Operations → Select. Formats missing period dates as 'MMM d, yyyy' for display in the email body.
16	Build Missing Dates List	Inside loop. Data Operations → Join. Joins formatted dates with HTML line break separator.
17	Has Missing Weeks	Inside loop. Control → Condition. Evaluates <code>length(Find Missing Periods) > 0</code> .
18	Send Reminder Email	Inside loop, True branch. Office 365 Outlook → Send email from shared mailbox. From: <code>school_leadership@crps.co.us</code> . To: member email. Cc: management emails. HTML body with missing week list and Power App link.

5.2 Connector Dependencies

Connector	Tier	Used For
Office 365 Groups	Standard	Listing security group members (steps 1–2)
SharePoint	Standard	Querying ReportingPeriods and Submissions lists (steps 9–10)
Office 365 Outlook	Standard	Sending email from shared mailbox (step 18)
Data Operations	Built-in	Select, Filter Array, Join, Compose actions throughout
Control	Built-in	Apply to Each loop and Condition action

i All connectors are standard tier. No premium Power Automate license is required to run these flows under Microsoft 365 Business Standard.

6. Operational Runbook

This section documents all recurring and on-demand maintenance procedures for the system.

6.1 Add a New Team Member

The team's manager and executive assistants are group owners and they independently manage group membership as the team evolves. If the team requests assistance onboarding a new team member, the steps are:

1. Confirm approval by the team manager for the request.
2. Add the user to **the `s1_sec_teammember`** security group. No modifications to the Power BI workspace or Power Apps sharing settings are required. All permissions flow through the security groups.

3. Test the Power App by having the new user open the app and verify they can submit and view their own data only.
4. Test the Power BI report by having the user open the app and navigate to the *Reports* screen. The first time the user views the report, Power BI will prompt the user to sign-in. Verify that the user can view all report visuals and that they can only see their own data.

6.2 Remove or Adjust a Team Member's role

The team's manager and executive assistants are group owners and they independently manage group membership as the team evolves.

If the team requests assistance offboarding a team member the steps are:

1. Confirm approval by the team manager for the request.
2. Remove the user from the **sl_sec_teammember** security group. No modifications to the Power BI workspace or Power Apps sharing settings are required. All permissions flow through the security groups.
3. Historical submission data is retained in SharePoint, do not delete it. It will remain in the model for prior-year reporting.

If the team requests assistance changing a team member's role, the steps are:

1. Confirm approval by the team manager for the request.
2. Add or remove the user from the **sl_sec_teammember** or **sl_sec_management** security groups as requested. Team members may sometimes be temporarily added to the Management security group for periods when they are filling the role of "acting manager." In this scenario, retain the user in the Team Member group, add them to the Management group, then remove them from the Team Member group when their acting term expires. No modifications to the Power BI workspace or Power Apps sharing settings are required. All permissions flow through the security groups.

6.3 Add a New School Year

Before the start of each school year - typically mid-late August, the **ReportingPeriods** SharePoint list must be updated with the upcoming school year's week-ending dates.

The team's manager and executive assistants have *Write* permission for the SharePoint lists and they independently manage updating the lists when needed.

If the team requests assistance updating the **ReportingPeriods** lists, the steps are:

1. Obtain from the team's manager or executive assistants, the list of reporting periods to be added. Consider sending them a spreadsheet in a format matching the SP list schema.
2. Edit the list in grid view and past the period data from the spreadsheet into the list. Exit grid view.

6.4 Update Flow Connector Credentials

If a connector connection expires or credentials change (e.g. password rotation):

1. In Power Automate, go to Data → Connections.
2. Find the affected connection (Office 365 Groups, SharePoint, or Office 365 Outlook).
3. Click the three-dot menu and select 'Edit' or 'Fix connection'.

4. Re-authenticate with the correct credentials.
5. Both reminder flows share the same connections, updating a connection fixes both flows simultaneously.
6. Run a manual test of one flow to confirm connectivity is restored.

7. Design Decision Log

This section documents significant design decisions made during the build, including the reasoning and alternatives considered. It is intended to prevent future developers from 'fixing' things that were intentionally designed a specific way.

Decision	Choice Made	Reasoning and Alternatives
RLS Architecture (Option A vs B)	Option A: filter applied to <code>Users</code> dimension table. Propagates to <code>Submissions</code> and <code>TA_LeaveData</code> via relationships.	Option B (direct fact table filters) was implemented first but caused <code>REMOVEFILTERS()</code> to fail for team average measures. Option A also enables the <code>Users</code> table as a clean person dimension for report slicers. Trade-off: <code>REMOVEFILTERS(Users[Email])</code> still cannot override RLS, so team averages are not available in the <code>TeamMember</code> role.
WeekBridge as hub table	A dedicated calculated Saturday-only table connects <code>Submissions</code> and <code>TA_LeaveData</code> , with <code>DateTable</code> providing calendar attributes.	An alternative was to use <code>DateTable</code> directly as the hub. <code>WeekBridge</code> was chosen because it produces exactly one row per reporting week, making visual aggregations and compliance counting natural. A <code>ReportingPeriods</code> table in the model was considered and rejected to avoid manual maintenance.
ILFT denominator: SUMX row-by-row vs SUM then subtract	<code>SUMX(TA_LeaveData, MAX(0, 40 - TOTAL_LEAVE_HOURS))</code> iterates row by row before summing.	<code>SUM</code> approach (<code>40*n - SUM(leave)</code>) would produce an incorrect result for multi-person selections because it subtracts combined leave from a combined 40-hour pool rather than computing each person's available hours independently.
Cumulative ILFT baseline: extracted supporting measure	<code>ILFT - Cumulative Baseline</code> is a hidden supporting measure called inside a <code>SUMX</code> over <code>TA_LeaveData</code> within the cumulative calculation.	Inline nested <code>SUMX</code> was the original approach. The extracted measure improves readability of <code>ILFT - Cumulative</code> without sacrificing correctness. The measure is hidden to prevent misuse in visuals.

Decision	Choice Made	Reasoning and Alternatives
TODAY() vs TODAY()-1 for cutoff dates	TODAY() is used consistently across all measures and WeekBridge columns.	TODAY()-1 was tested earlier in development to include the current in-progress week on Fridays. This was reverted because it created inconsistencies between the compliance cards and the dashboard. The current reporting week logic (only include a Saturday once it has arrived) was instead implemented in [Current Reporting Week] and applied consistently.
Users table via Power Query (not DAX calculated table)	Users is built in Power Query as a UNION of distinct emails from Submissions and TA_LeaveData.	A DAX calculated table would shift under filter context during measure evaluation, making it unreliable as a stable dimension. Power Query tables are computed at refresh time and are immune to DAX filter context.
DateTable start date: September 1, 2024	DateTable begins September 1, 2024, aligning with the first complete school year in the model.	The original start date (January 1, 2024) caused the 2023-2024 school year to appear partially in WeekBridge with incorrect SchoolWeekNumbers. The boundary was moved to the first complete school year to ensure data consistency.
Power Automate: two standalone flows instead of child flow	LTT - Monday Reminder and LTT - Tuesday-Friday Reminder each contain the full flow logic.	A parent/child flow architecture was designed and partially implemented. The child flow (built in legacy designer) could not be registered as a valid child flow target by the parent flows (built in new designer) due to a designer compatibility constraint. Given the minimal duplication (only the recurrence trigger differs), two standalone flows were accepted as the pragmatic solution.
Power BI embed in Power App via TileUrl	Power BI Tile control with AllowNewAPI = true and TileUrl pointing to the reportEmbed URL.	The native Power BI Tile control only supports dashboard tiles, not full report pages. The TileUrl approach with AllowNewAPI enables full interactive report embedding without a premium Power Apps license.

8. Known Limitations and Backlog

8.1 Known Limitations

Limitation	Detail and Workaround
Calculated table staleness	WeekBridge columns using TODAY() (IsFutureWeek, IsCurrentSchoolYear, IsCurrentReportingWeek) are computed at refresh time. Between refreshes, these flags may be stale by up to 24 hours. Impact is limited to visual-level filters; all measures compute their own real-time cutoff logic. Workaround: refresh daily.
Team averages unavailable in TeamMember role	ILRP - Team Avg and ILFT - Team Avg cannot be used in TeamMember report views due to RLS preventing REMOVEFILTERS() from lifting the person filter. These measures function correctly in the Management role. Workaround: none without pre-computing averages outside the RLS-filtered data model.
Power BI Desktop View As not reliable for composite models	The View As Role feature in Power BI Desktop does not correctly filter TA_LeaveData in a composite model. Always test RLS by publishing to the Service and viewing as the target user.
Power Automate flows are duplicated	The Monday and Tuesday-Friday flows contain identical logic due to a child flow compatibility constraint. Any logic change must be applied to both flows. Risk: flows drift out of sync. Mitigation: maintain a comment in each flow noting that the other flow must be updated simultaneously.
ReportingPeriods requires annual maintenance	The ReportingPeriods SharePoint list must be extended with new Saturday dates at the start of each school year. This is the only required annual maintenance task. See Section 6.3 .

8.2 Backlog Items

#	Item	Priority	Description
1	Power App: lock Submission screen for Management users	High	Users in <code>s1_sec_management</code> should not be able to submit time entries. Add a security group check on <code>scrSubmit.OnVisible</code> to redirect management users to <code>scrReports</code> .
2	Power App: 'On behalf of' submission for managers	Medium	Allow managers and admin assistants to submit entries on behalf of team members. Would require a team member selector on <code>scrSubmit</code> and modified submission logic to populate <code>SubmittedByEmail</code> from the selection rather than the current user.
4	Child flow architecture for Power Automate	Low	Revisit the parent/child flow pattern once the legacy designer compatibility constraint is resolved in Power Automate. Would eliminate the duplication risk between the two reminder flows. Explore implementing <i>connection references</i> to enable this architecture.
6	Role-based bookmark loading in Power App	Low	Investigate whether the Power BI embed <code>TileUrl</code> supports a <code>bookmarkGuid</code> parameter that can be set dynamically based on the user's role. Would allow managers to load the Manager View automatically without clicking the in-report button.
7	Power BI: Replace WeekBridge and DateTable tables with ReportingPeriods table	Low	Since the <code>ReportingPeriods</code> SP list must be maintained for the other solution components (Power App, Power Automate flow), replace the current <code>WeekBridge</code> and <code>DateTable</code> tables in the Power BI semantic model with a <code>ReportingPeriods</code> table pulled from the SharePoint list (adding calculated columns if needed). This will require remediation of most measures and visuals however it will make the <code>ReportingPeriods</code> list the “single source of truth” for the reporting periods dimension.
8	Power BI: Implement team average measures	Medium	Consider creating a <code>TeamAverages</code> table in Power Query that provided anonymized team average metrics that can be used in the data model without being restricted by RLS imitations or violating the user privacy requirements enforced by RLS.

8.3 Document Maintenance

This document should be reviewed and updated when:

- Any backlog item is implemented
- A known limitation is resolved
- Security group membership or Power BI roles change structurally
- New measures are added to the semantic model

Maintained by the implementing contractor. Version history should be tracked in the document filename (e.g. `TechRef_v1.1`).