

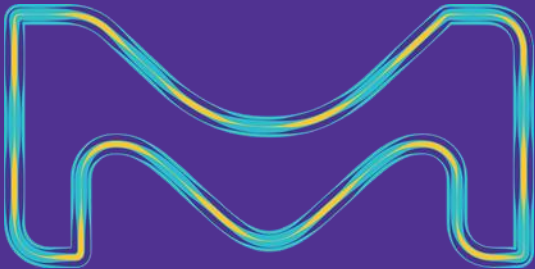
Elevating Data Advocacy

Transforming Dashboards with React.JS and AI-Powered Innovation

Richard Becker, London UK, Merck

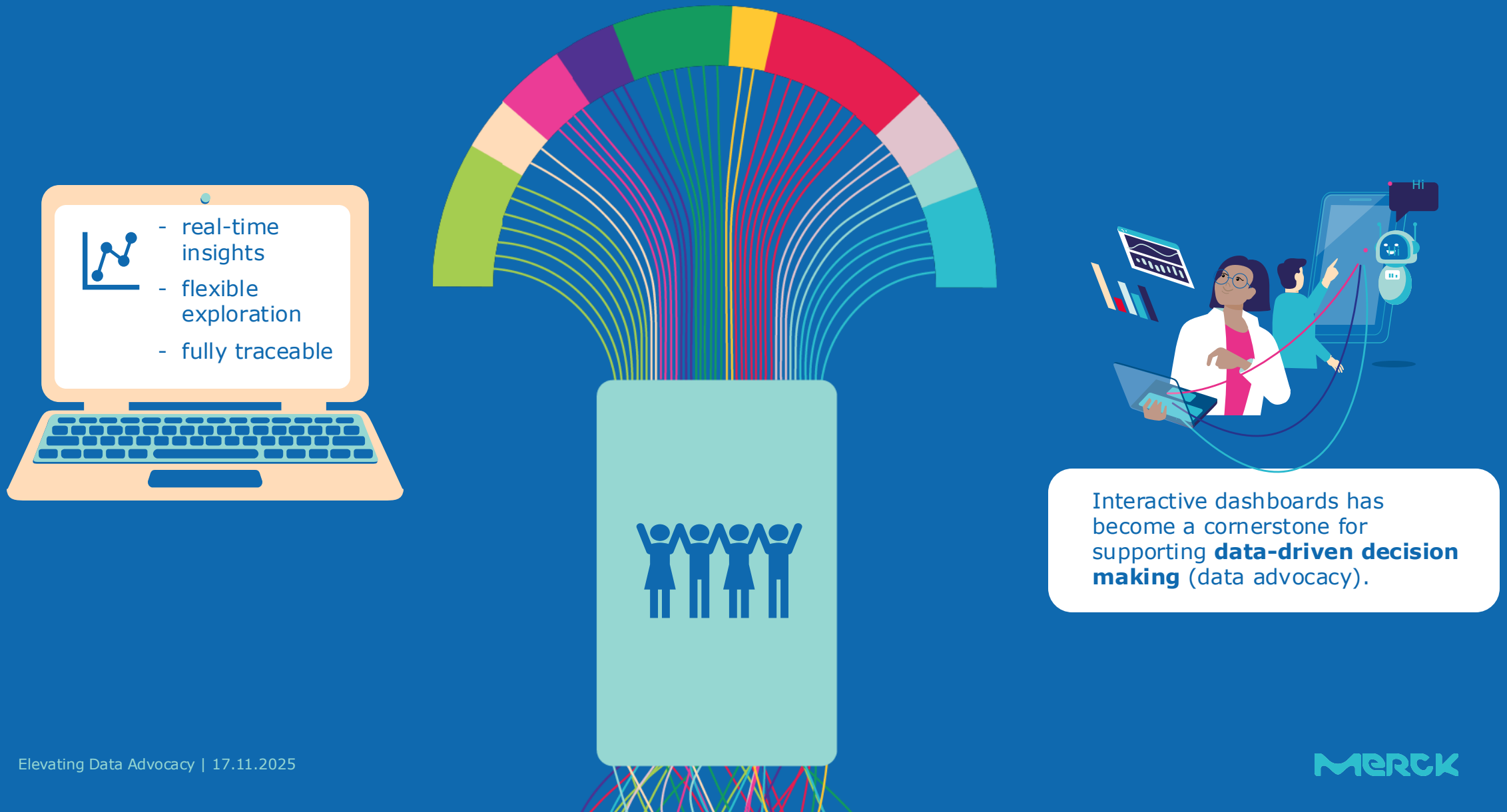
Markus Niederstrasser, Tokyo Japan, Merck

PHUSE EU Connect 2025



MERCK

Background and Motivation



Agenda

- 1 **dashboard framework**
- 2 **React JS**
- 3 **LLM**

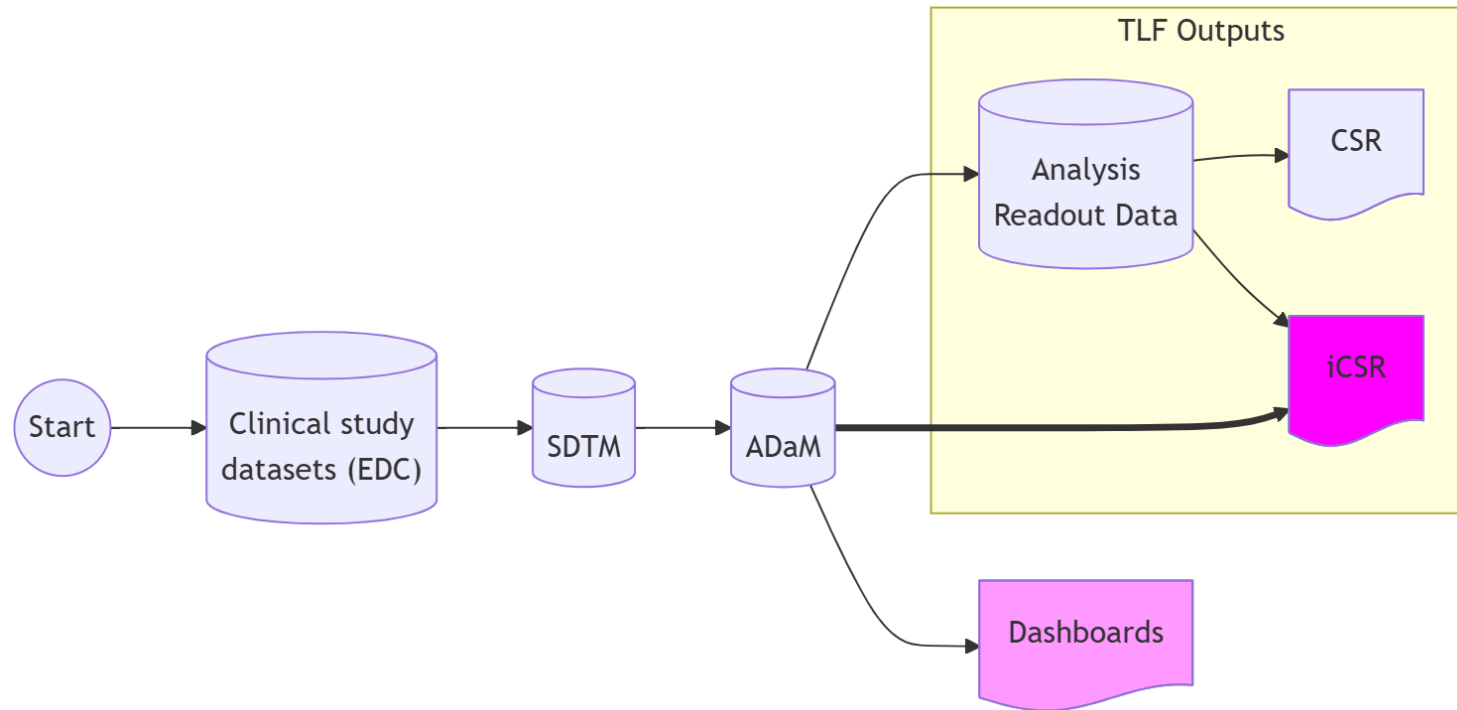
dashboard framework

1

System architecture

Overview in the APS E2E pipeline

The APS E2E pipeline outlines the data flow and the components involved in output creation as detailed below.



iCSR

Motivation

The iCSR is an interactive **R Shiny application** designed to facilitate the generation of Clinical Study Report **outputs**.

Its primary goal is to move closer to the objective of achieving “**Keystats within a day.**”

The aim is to develop an **interactive, traceable, and adaptable** iCSR tool that prioritizes visualization, **real-time data manipulation**, and **regulatory compliance**, while also enabling straightforward adaptation and interpretation by subject matter experts (SMEs).

Mock Table ID	CSR TLF No.	TLF Title	CSR Mandatory for Onc. Studies	CSR Mandatory for Non Onc. Studies	Key Stats
T.DISPI or T.DISPD	8.1.1.1	Subject Disposition Status	Yes	Yes	x
T.ANASET	8.1.1.3	Analysis Sets	Yes	Yes	x
T.DMCHAR	8.1.4.1	Demographic Characteristics - Full Analysis Set	Yes	Yes	x
T.DISHIST	8.1.6.1	Disease History - Full Analysis Set	Yes	Yes	x
T.EXDUR	8.1.8.1	<Drug name> - Duration of Therapy [weeks] - SAF Analysis Set	Yes	Yes	x
T.EXDOSE	8.1.8.3	<Drug name> - Cumulative Dose, Dose Intensity, Relative Dose Intensity - SAF Analysis Set	Yes	Yes	x
T.EFFOS1	8.2.1.1	Primary Analysis - Full Analysis Set	Yes	No	x
F.EFFKM	8.2.1.4	Kaplan Meier Curve - Full Analysis Set	Yes	No	x
F.FORESTOS	8.2.1.7	Forest Plot of Hazard Ratio by Subgroups - Full Analysis Set	Yes	No	x
T.EFFPFS1	8.2.2.1	Primary Analysis - Full Analysis Set	Yes	No	x
F.EFFKM	8.2.2.4	Kaplan Meier Curve - Full Analysis Set	Yes	No	x
F.FORESTOS	8.2.2.7	Forest Plot of Hazard Ratios by Subgroups - Full Analysis Set	Yes	No	x
T.EFFBOR1	8.2.3.1	Objective Response - Full Analysis Set	Yes	No	x
F.EFFFORESTBOR	8.2.3.5	Odds Ratios by Subgroups - Full Analysis Set	Yes	No	x
T.TEAEIOV	8.3.1.1	Overview of Treatment Emergent Adverse Events (TEAEs) - SAF Analysis Set	Yes	Yes	x
T.TEAEIOV	8.3.1.3	TEAEs by Primary System Organ Class (SOC) and Preferred Term (PT) - SAF Analysis Set	Yes	Yes	x
T.TIER1	8.3.1.4	Tier 1/3 Pre-specified TEAEs - SAF Analysis Set	Yes, for randomized studies with sample size ≥200 subjects	Yes, for randomized studies with sample size ≥200 subjects	x
T.TEAEIOV	8.3.1.9	Study Intervention Related TEAEs by SOC and PT - SAF Analysis Set	Yes	Yes	x
T.TEAEIOV	8.3.1.13	Serious TEAEs by SOC and PT - SAF Analysis Set	Yes	Yes	x
T.TEAEIOV	8.3.1.15	Study Intervention Related Serious TEAEs by SOC and PT - SAF Analysis Set	Yes	Yes	x
T.TIER1EIR	8.3.1.16	Exposure Adjusted Incidence Rates for Tier 1/3 prespecified TEAEs - SAF Analysis Set	Yes, for randomized studies with sample size ≥200 subjects and treatment exposure varying between groups	Yes, for randomized studies with sample size ≥200 subjects and treatment exposure varying between groups	x
T.LABSHIFW	8.3.3.6	Hematology - Shift from Baseline to Worst On-treatment Value - SAF Analysis Set	Yes	Yes	x
T.LABSHIFW	8.3.3.16	Biochemistry - Shift from Baseline to Worst On-treatment Value - SAF Analysis Set	Yes	Yes	x

Protocol: <XXXXX> (shortname) - Cutoff date: DDMMYYYY (<Purpose>)

T.DISPD

8.1.1: Study Subject Data

8.1.1.1: Subject Disposition

Table 8.1.1.1: Subject Disposition Status (version: study completion is unequal to treatment completion)

	Treatment A	Treatment B	Total
Number of screened subjects, n			XX
Subjects discontinued prior to randomization, n			XX
Subject did not meet all eligibility criteria			XX
Withdrawal by Subject			XX
Other			XX
Randomized subjects, n (%)	XX (100.0)	XX (100.0)	XX (100.0)
End of treatment status <Treatment XXX>			
Received no treatment (a), n (%)	XX (XX.X)	XX (XX.X)	XX (XX.X)
Treatment ongoing, n (%)	XX (XX.X)	XX (XX.X)	XX (XX.X)
Treatment completed (b), n (%)			
X Cycles received	XX (XX.X)	XX (XX.X)	XX (XX.X)
Progressive Disease	XX (XX.X)	XX (XX.X)	XX (XX.X)
Death	XX (XX.X)	XX (XX.X)	XX (XX.X)
Treatment discontinued, n (%)			
Primary Reason:			
Adverse Event	XX (XX.X)	XX (XX.X)	XX (XX.X)
Lost to Follow-up	XX (XX.X)	XX (XX.X)	XX (XX.X)
Withdrawal by Subject	XX (XX.X)	XX (XX.X)	XX (XX.X)
Protocol Non-Compliance	XX (XX.X)	XX (XX.X)	XX (XX.X)
xxx	XX (XX.X)	XX (XX.X)	XX (XX.X)

Source: ADSL DDMMYYYY hh:mm <creation date of ADAM dataset>; Listing 9.XX; SDTM package: DDMMYYYY

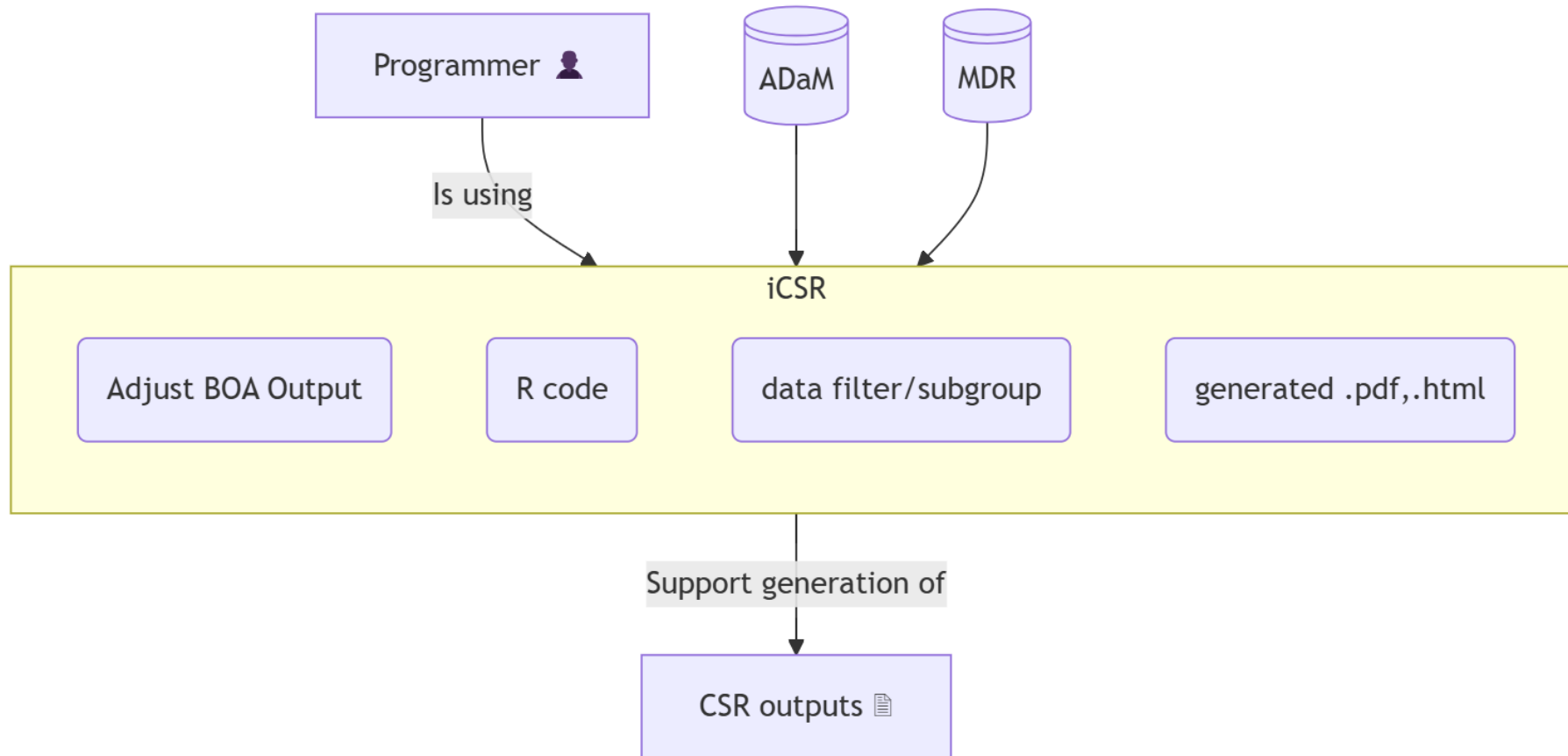
Program_path\pgm_name.sas, DDMMYYYY hh:mm

Page 1 of x

iCSR

Common use case

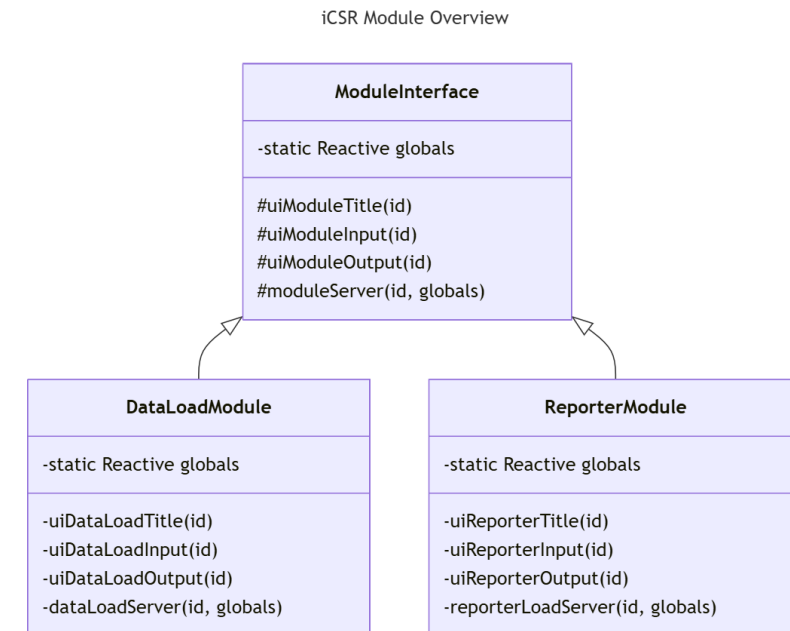
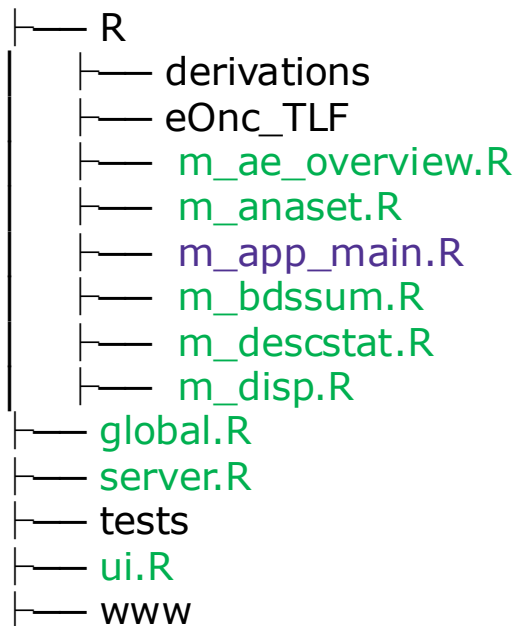
Stat/APS: Creation of Keystats



iCSR

App - Modules and interactions

Modular setup

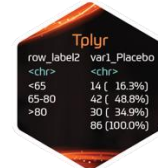
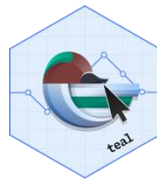


Each module has its own namespace and defines UI and Server components. The UI and Server code is within one .R file (e.g. m_demog.R)

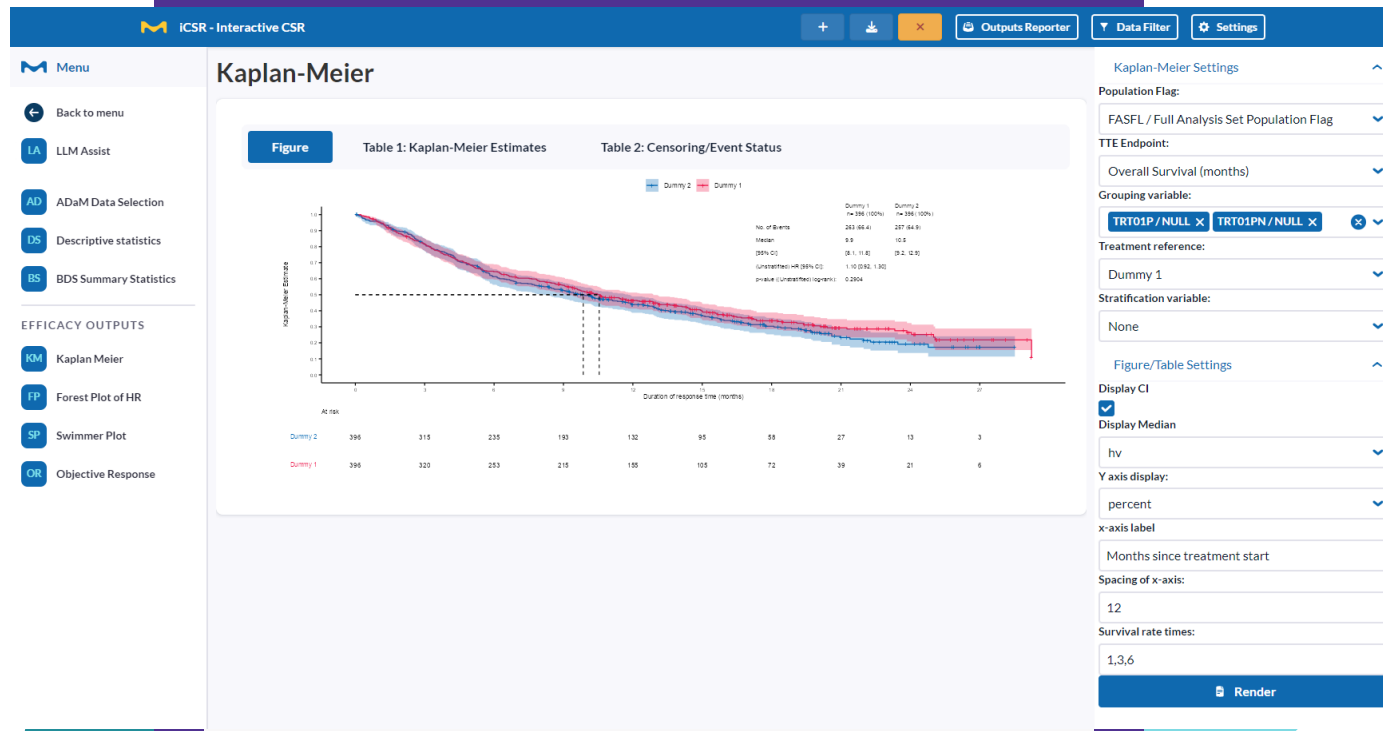
The reactive variable “globals” is shared between modules and facilitates exchange of loaded data(ADaMs), outputs, settings.

App - R packages used / framework stack

- **R Shiny**: Ground framework.
- **Liquid.react**: React.JS components following Merck Liquid Design pattern.
- **Tplyr**: Used within RDOT function library for output creation.
- **teal.reporter**: „Basket“ for report creation. The actual renderer is coming from RDOT.
- **teal.filter/data**: Data filter panel
- **shinymeta** (instead of teal.code): to capture logic in a Shiny app and exposing it as code.
- **Shinytest2**: unit and integration tests using testthat and chromote (headless Chrome browser)
- **gt**: output creation (.rtf, .pdf)



iCSR example



Key Features:

- **CSR keystate**;
Descriptive statistics/frequency; BDS summary stat
- R code generation for **RDOT**
- “**Basket** feature” for generated outputs
- BOA compliant **output generator** (.pdf, .html)
- Data **filter** by domain
- ADaM **codelist import**
- Storage Layer ADaM access*
- Subgroup definition*

Testing strategy

Unit and Integration Tests

Unit Tests



Scope

- test small, isolated functions and module server logic



Speed

- fast, deterministic runs with minimal dependencies using fixtures/mocks



Purpose

- catch logic errors early, enable safe refactoring, and document intent



Tooling

- testthat
- Partial dummy data

Integration Tests



Scope

- validate interactions between modules and end-to-end UI flows



Speed / Environment

- app-level/browser-driven tests with complete dummy data and shared state



Purpose

- verify wiring, reactivity, and user paths across components
- Actual use case tests



Tooling

- shinytest2 + Chromote
- use explicit waits/selectors and regression snapshots

React JS

2

Merck Liquid Oxygen

Award winning Merck owned UX Design System

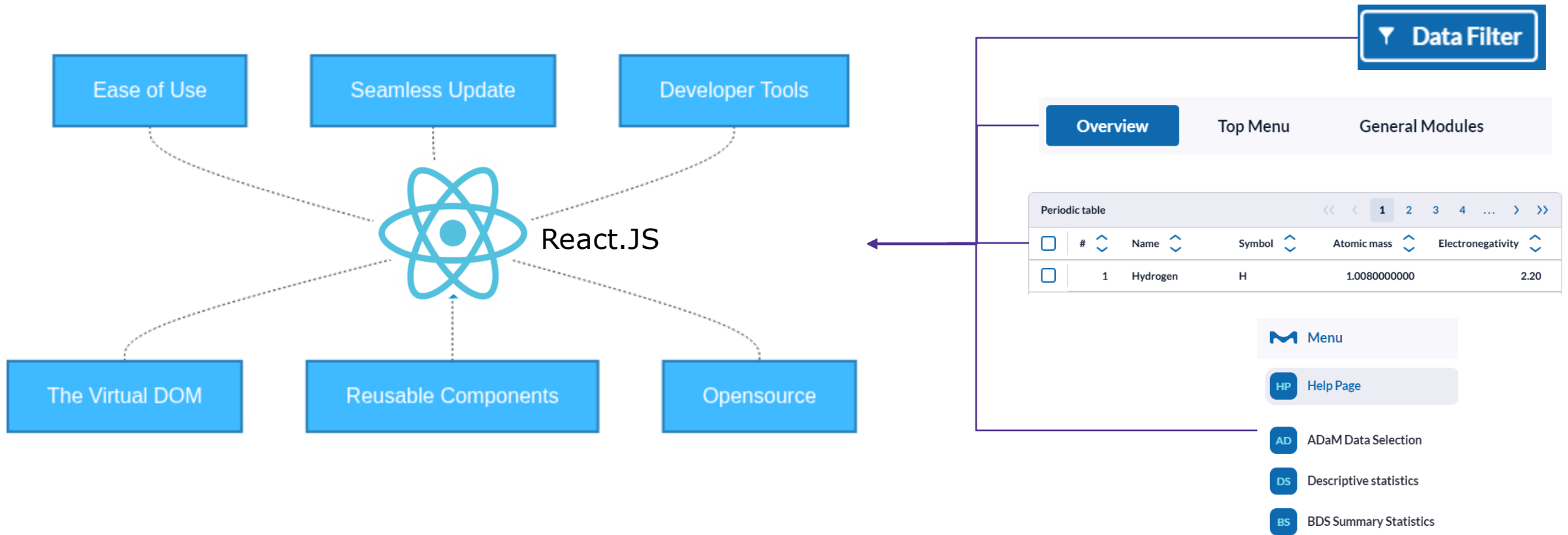


Why use Merck Liquid Oxygen

- Design Compliance, consistent look and feel
- Accessibility
- Framework agnostic (React.JS, Vue.JS,...)
- Well documented
- Written in Typescript
- Small footprint

React.JS

React.JS is a declarative, component-based JavaScript library for building fast, accessible user interfaces that scale from simple widgets to complex applications.



Liquid.react

In house R package {liquid.react} that capsules the Merck Liquid Design React.JS components for use in R Shiny. Python version is also available.



JS wrapper
@/shiny.react



- 1) setup index.js
- 2) > npx webpack



```
library(liquid.react)  
  
LdHeader()  
LdButton()  
LdIcon(name='loading')
```

LLM

3

LLM Integration

How to integrate LLM to Shiny dashboards?

Server

{ellmer} + toolcalling

UI

{shinyChat}

- ellmer chat loaded within `moduleServer()`
- toolcalling allows the LLM to interact with app.
 - extend LLM with external tools (API calls,...)
- Control UI: update filter,...
- Update reactive values, state of app

What's the current temperature in Duluth, MN?



I'll check the current temperature in Duluth, Minnesota for you.

View result from `get_current_weather` ▶

The current temperature in Duluth, MN is 4.1°C (about 39.4°F) with wind speeds of 15.0 km/h.

Enter a message...



LLM Integration

What kind of LLM use cases are useful for dashboards?

- Chat assistant for navigation and in-context documentation, tailored to the current module and data view.
- AI-generated summaries or translations of figures/tables to accelerate drafting of CSR-ready narratives.
- Dashboard control via tool calling (e.g., apply filters, switch modules, preconfigure inputs) when workflows are complex or repeated.
- R code generation leveraging an internal snippet repository to scaffold reproducible filters, subgroup definitions, or output customizations.

LLM Integration

Implemented use cases

- R code generator for dynamic subgroups
- Documentation helper and navigation assistant

The screenshot displays the iCSR - Interactive CSR web application. The interface includes a top navigation bar with a logo, the title "iCSR - Interactive CSR", and buttons for adding, downloading, deleting, reporting outputs, filtering data, and settings. A left sidebar menu contains options like "Menu", "Help Page", "LLM Assist", "ADaM Data Selection", "Descriptive statistics", "BDS Summary Statistics", and "TABLES, LISTINGS AND FIGURES" with sub-options "Safety" and "Efficacy". The main content area is titled "LLM Assistant Chat". It shows a user prompt: "Create a new subgroup variable named ADSL.AAGEGR2 based on ADSL.AAGE using the criteria <75;>=75". The assistant's response explains the process and provides the following R code:

```
# Create new age group variable AAGEGR2 based on AAGE
ADSL <- ADSL %>%
  mutate(AAGEGR2 = case_when(
    AAGE < 75 ~ "<75",
    AAGE >= 75 ~ ">=75",
    TRUE ~ NA_character_
  ))
```

Below the code, the assistant explains that this R code will:

1. Use the existing **AAGE** variable (analysis age) from the ADSL dataset
2. Create a new character variable **AAGEGR2** with two categories:
 - "<75" for subjects with age less than 75 years
 - ">=75" for subjects with age greater than or equal to 75 years
3. Handle any missing values in **AAGE** by assigning **NA** to the new variable

The assistant further explains that the `case_when()` function from dplyr provides a clean way to create categorical variables based on multiple conditions, and the `TRUE ~ NA_character_` ensures that any cases not covered by the first two conditions (such as missing AAGE values) are assigned NA.

At the bottom of the chat interface is a text input field labeled "Enter a message..." and a send button.

Where to go from here?

Next steps

- Use of Meta Data Repository (MDR)
- In-house R Shiny dashboard framework
- LLM coding assistant with access to internal tools and frameworks

Next milestone features for iCSR

- More standard outputs (PK, EAIR, ...)
- Bookmarks, save/restore dashboards state.
- Use of LLM agents to automate routine tasks
 - Ingest a Programming Plan, extract analysis requirements, load data/codelists (MDR).
- Enhance integration and automation into the APS end-to-end pipeline, by exposing iCSR functionality via API.



The background is a solid blue color. On the left side, there are several large, overlapping yellow shapes with rounded corners and edges, creating a dynamic, abstract composition. One large yellow shape starts from the top left and extends towards the center. Another yellow shape is positioned above it, and a third is below it. A fourth yellow shape is located further down and to the right.

Thank you
for your attention

Backup

iClass

Future in-house framework

limitations with existing frameworks:

- Support for subgroup comparison is limited with existing framework.
- „individual“ vs „group mode“ concept as a must have feature
- Reporting needs update to support internal pipeline/packages {RDOT} and support to deferring rendering at ‚checkout‘ rather than ‚Add to basket‘.

				
Modules	Unified module configuration	✓	✓	✓
	Group-level modules	✓	✓	✓
	Nested group-level modules (domains)	✓		✓
	Individual-level modules			✓
	Nested individual-level modules			✓
Global Concepts	Focal dataset (global filters)	✓	✓	✓
	(Nested) subgroup comparisons			✓
	Patient selection (shared across modules)			✓
	Patient flagging			✓
Reporting	Group-level reporting	✓		✓
	Individual-level reporting			✓
	Batch individual-level reporting			✓
	Batch domain-level reporting			✓