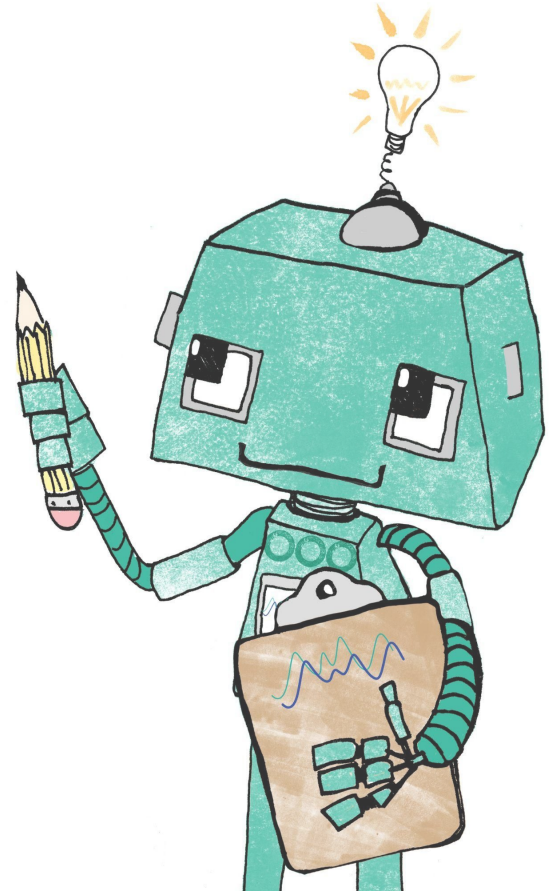


Scaling R with AWS Lambda

Jack Walton | Jumping Rivers



github.com/nationalarchives/DiAGRAM



Jumping Rivers

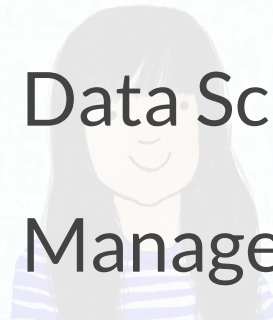
R / Python Consultancy

Data Science / ML

Managed RStudio Services

Security Audits

Training



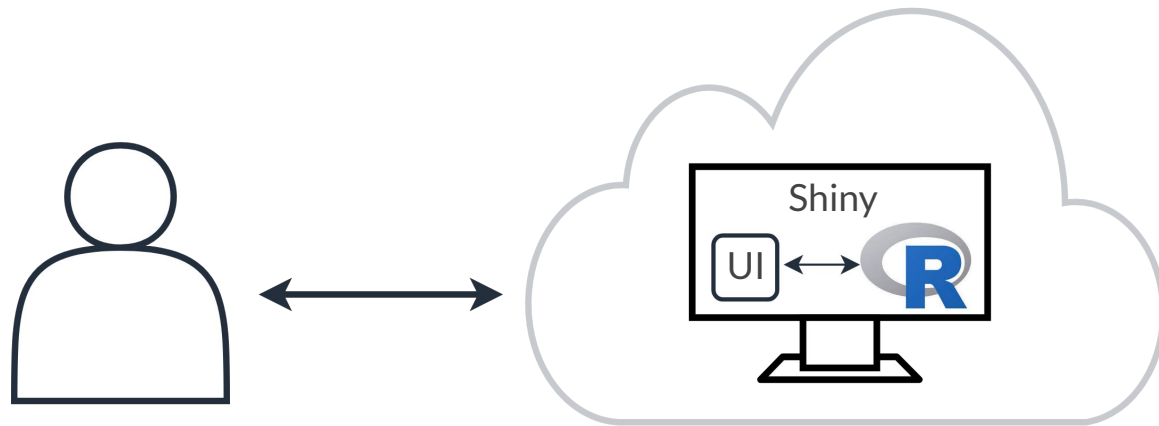
Evolution of a client project

Phase 0: Shiny



Evolution of a client project

Phase 0: Shiny



Evolution of a client project

Phase 0: Shiny

Shiny application which takes input from a user and can:

Evolution of a client project

Phase 0: Shiny



Shiny application which takes input from a user and can:

 Evaluate a Bayesian network model

Evolution of a client project

Phase 0: Shiny




Shiny application which takes input from a user and can:

-  Evaluate a Bayesian network model
-  Produce PDF reports via R Markdown

Evolution of a client project

Phase 0: Shiny

Shiny application which takes input from a user and can:

-  Evaluate a Bayesian network model
-  Produce PDF reports via R Markdown
-  Produce summary plots via `{ggplot2}`

Evolution of a client project

Phase 0: Shiny

Evolution of a client project

Phase 0: Shiny

 Struggling to meet accessibility requirements w/ shiny

Evolution of a client project

Phase 0: Shiny

 Struggling to meet accessibility requirements w/ shiny

 Solution lives separate to *all* other client inf
(shinyapps.io vs. AWS)

Evolution of a client project

Phase 0: Shiny

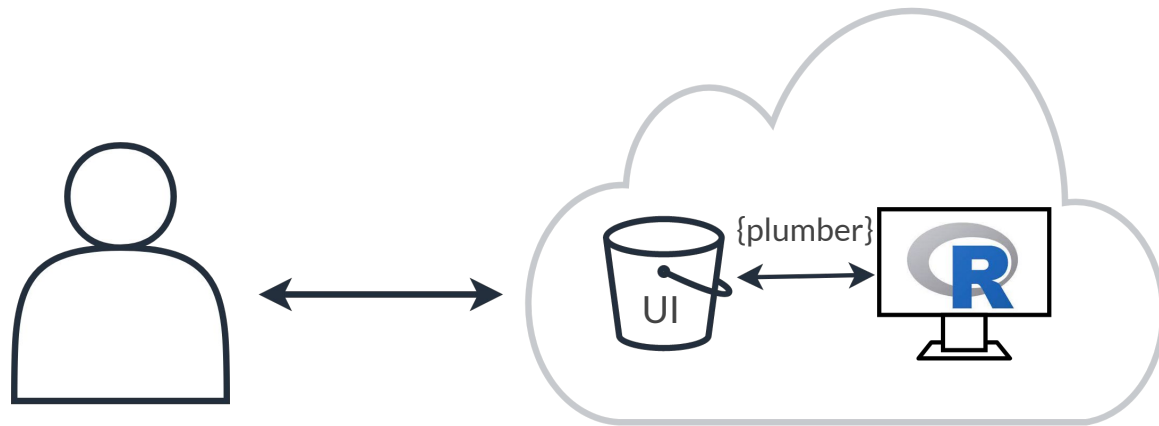
- 🙄 Struggling to meet accessibility requirements w/ shiny
- 😓 Solution lives separate to *all* other client inf
(shinyapps.io vs. AWS)
- 🤔 Uncertainty on how *best* to scale

Evolution of a client project

Phase 1: Shiny → JavaScript + {plumber}

Evolution of a client project

Phase 1: Shiny → JavaScript + {plumber}



Evolution of a client project

Phase 1: Shiny → JavaScript + {plumber}

Evolution of a client project

Phase 1: Shiny → JavaScript + {plumber}



Accessibility requirements met with JS

Evolution of a client project

Phase 1: Shiny → JavaScript + {plumber}



Accessibility requirements met with JS



Solution (bucket + EC2) moved to live with rest of inf

Evolution of a client project

Phase 1: Shiny → JavaScript + {plumber}



Accessibility requirements met with JS



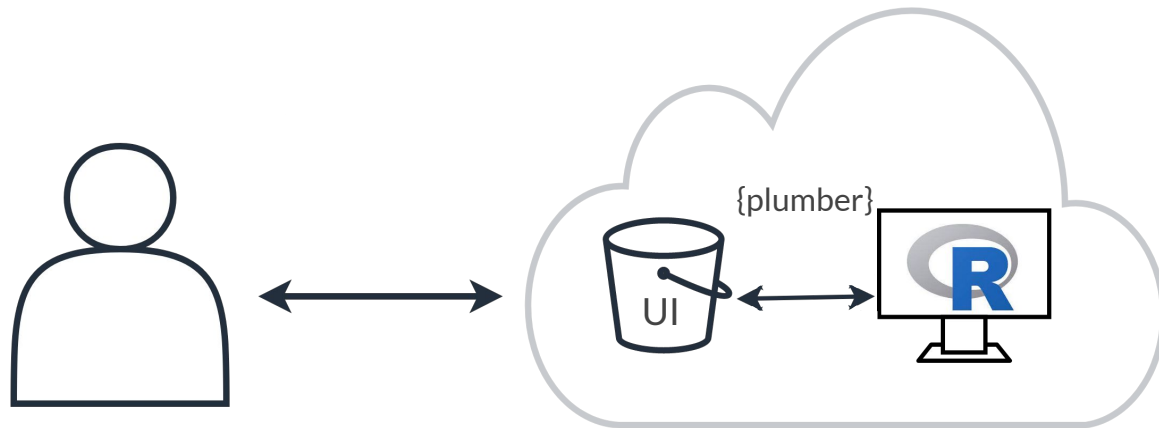
Solution (bucket + EC2) moved to live with rest of inf



Opened up options for scaling

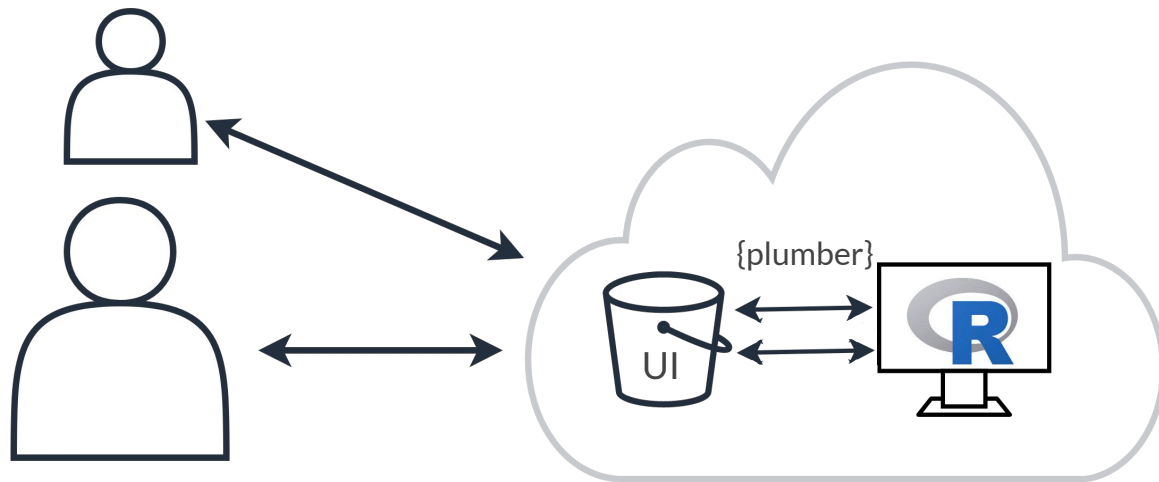
Evolution of a client project

Phase 1: Shiny → JavaScript + {plumber}



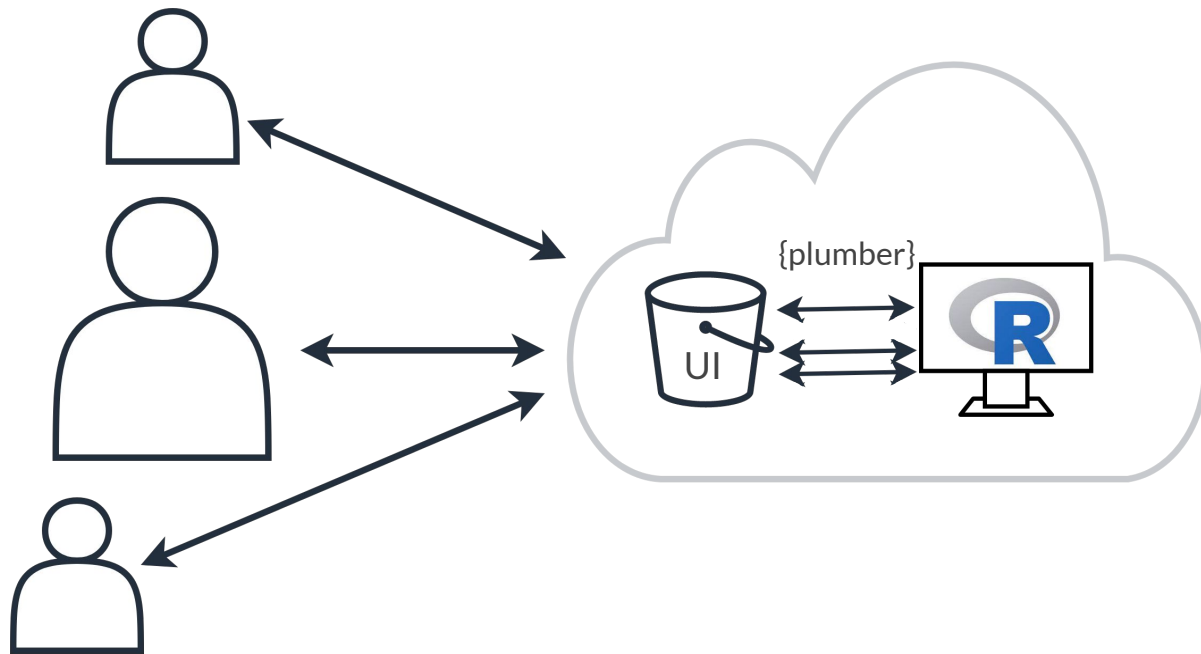
Evolution of a client project

Phase 1: Shiny → JavaScript + {plumber}



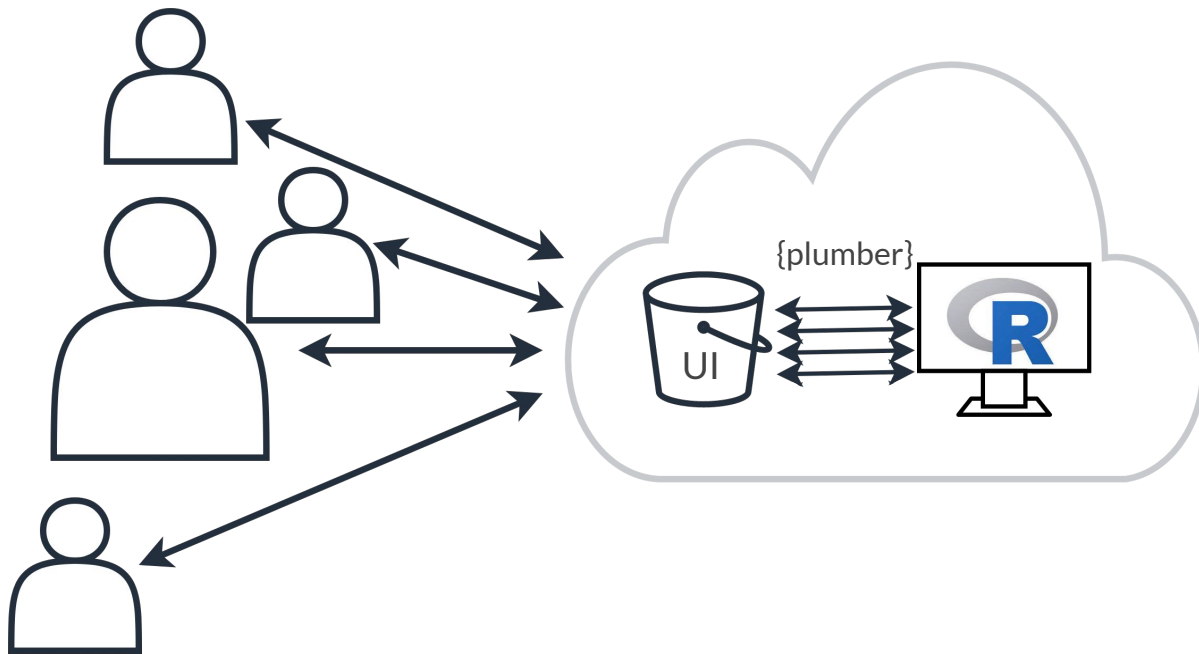
Evolution of a client project

Phase 1: Shiny → JavaScript + {plumber}



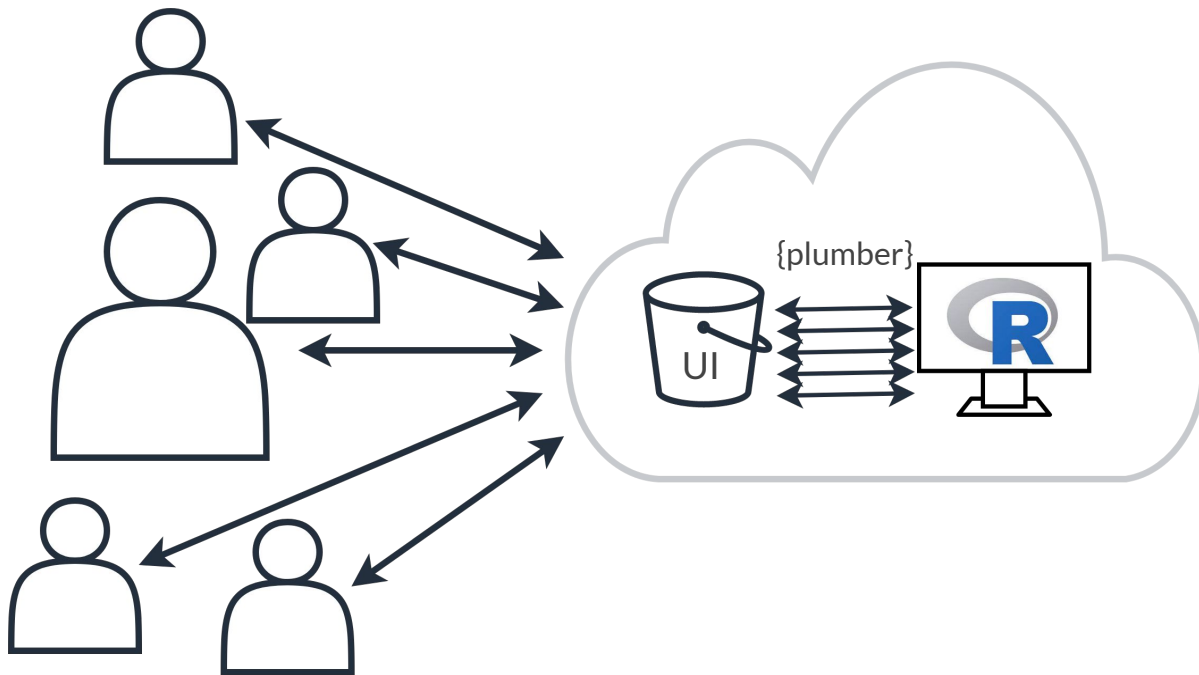
Evolution of a client project

Phase 1: Shiny → JavaScript + {plumber}



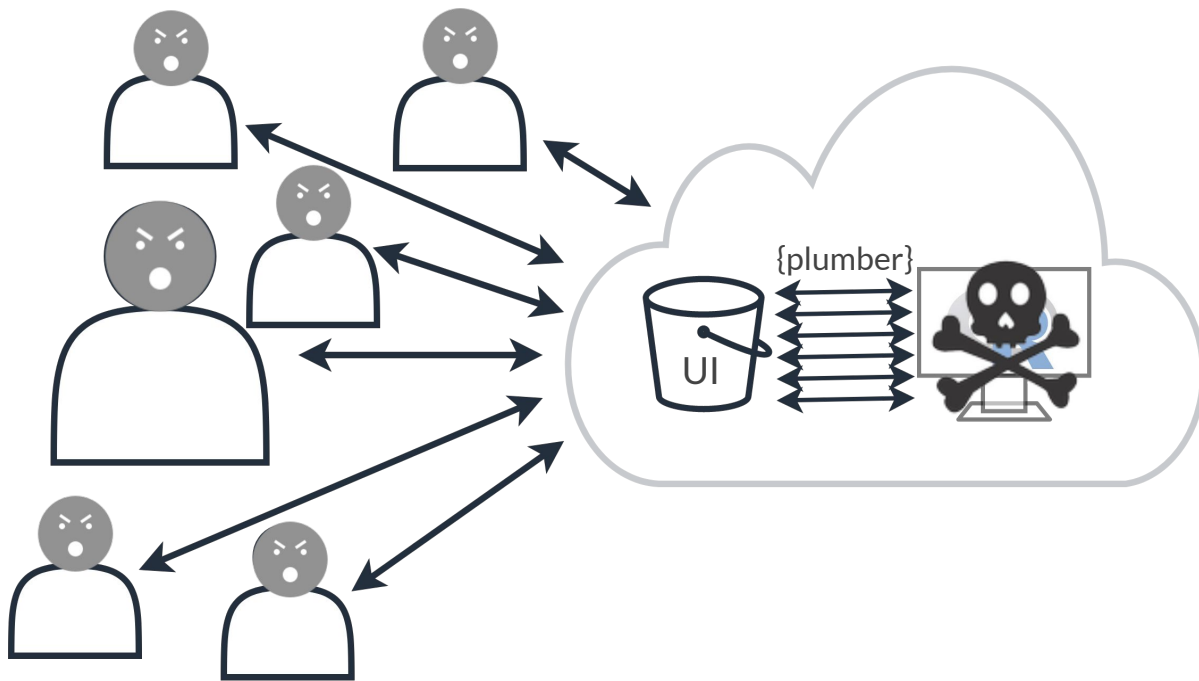
Evolution of a client project

Phase 1: Shiny → JavaScript + {plumber}



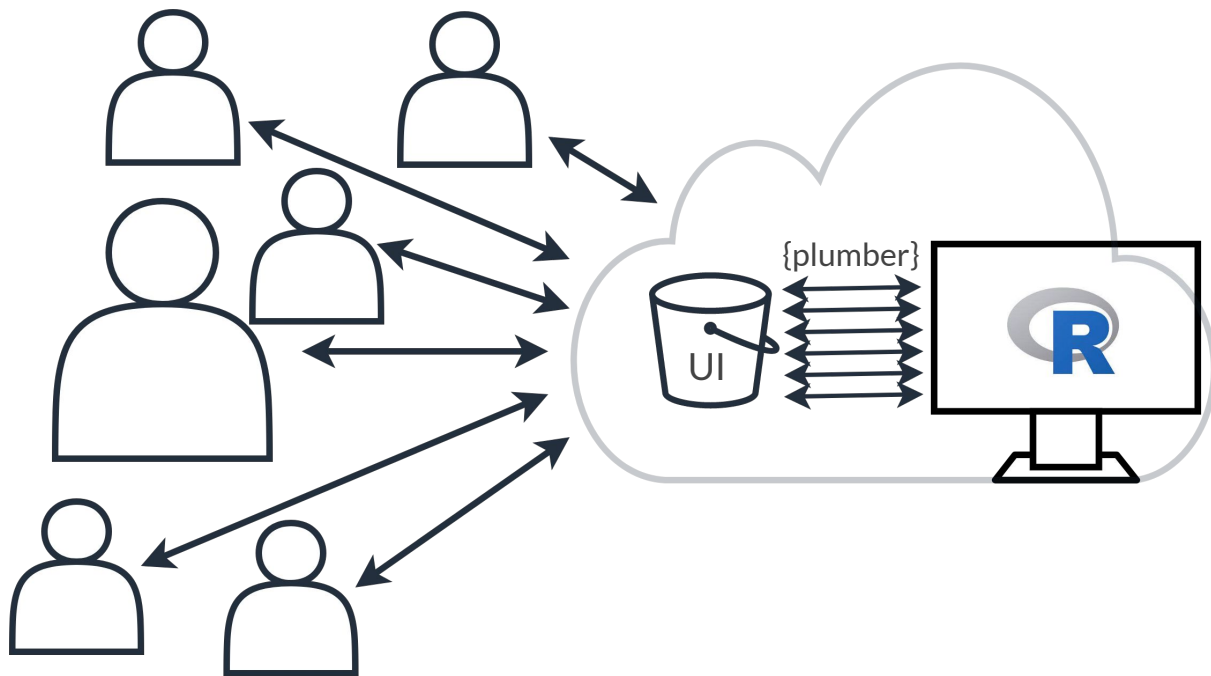
Evolution of a client project

Phase 1: Shiny → JavaScript + {plumber}



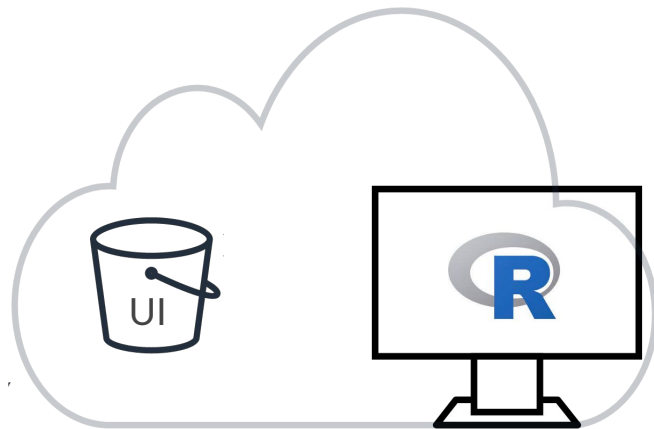
Evolution of a client project

Phase 1: Shiny → JavaScript + {plumber}



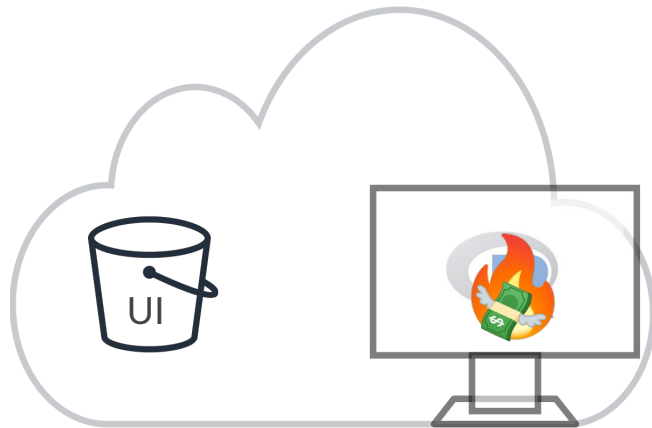
Evolution of a client project

Phase 1: Shiny → JavaScript + {plumber}



Evolution of a client project

Phase 1: Shiny → JavaScript + {plumber}



Evolution of a client project

Phase 2: JavaScript + {plumber} → Javascript + ???

Would like compute that:

Evolution of a client project

Phase 2: JavaScript + {plumber} → Javascript + ???

Would like compute that:



Scales based on demand

Evolution of a client project

Phase 2: JavaScript + {plumber} → Javascript + ???

Would like compute that:

 Scales based on demand

 Doesn't cost money when not in use

Evolution of a client project

Phase 2: JavaScript + {plumber} → Javascript + ???

Would like compute that:



Scales based on demand



Doesn't cost money when not in use



Can respond to requests quickly

Introducing: Functions as a Service

FaaS, offered by multiple cloud providers:

- ✓ Scale based on demand
- ✓ Don't cost money when not in use
- ✓ Can respond to requests quickly

Introducing: Functions as a Service

FaaS, offered by multiple cloud providers:

- ✓ Scale based on demand
- ✓ Don't cost money when not in use
- ✓ Can respond to requests quickly
- 💕 Are "Serverless", so require minimal maintenance


AWS Lambda + R


AWS Lambda + R



AWS Lambda supports a number of different “runtimes”


AWS Lambda + R

 AWS Lambda supports a number of different “runtimes”

 Runtimes represent the languages which can be used "out-of-the-box" in Lambda...


AWS Lambda + R


 AWS Lambda supports a number of different “runtimes”

 Runtimes represent the languages which can be used "out-of-the-box" in Lambda...

 ...R is not one of these

AWS Lambda + R

 AWS Lambda supports a number of different “runtimes”

 Runtimes represent the languages which can be used "out-of-the-box" in Lambda...

 ...R is not one of these...

 ...But Lambda supports “custom runtimes”:

 {lambdr} R package

 Docker

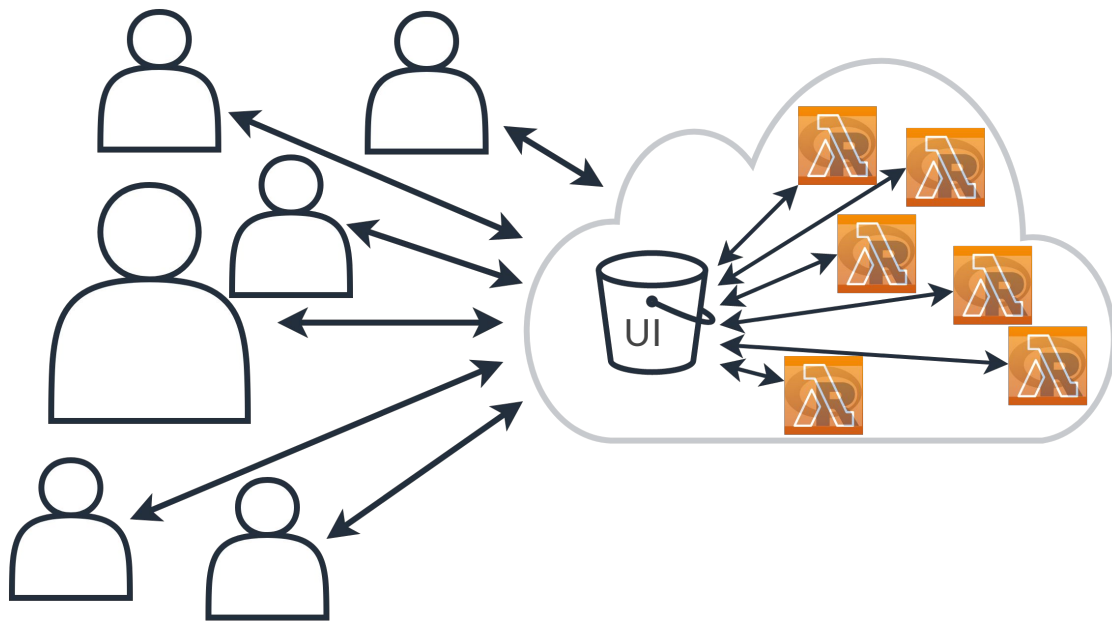
Evolution of a client project

Phase 2: JavaScript + {plumber} → JavaScript + AWS Lambda



Evolution of a client project

Phase 2: JavaScript + {plumber} → JavaScript + AWS Lambda



Evolution of a client project

Outcome of phase 1 & 2



WCAG 2.1 compliant frontend



Auto-scaling backend



Infrastructure as Code (terraform)



Automated deployments with GitHub Actions

Evolution of a client project

Sources & next steps

Frontend and Backend code already public:

github.com/nationalarchives/DiAGRAM

Infrastructure as code to be made public *soon*:

github.com/nationalarchives/DiAGRAM-terraform

Full application to go live in coming weeks:

diagram.nationalarchives.gov.uk

Evolution of a client project

Funding and contributors

Core JR contributors:

Tim Brock (JS & Accessibility; phase 1)

Jamie Owen (Shiny & {plumber}; phase 1)

Me (AWS Lambda & GHA & IaC; phase 2)



Thank you