

# R as a Service

11<sup>th</sup> July 2016



**Mark Sellors**

Head of Mango Data Labs

[msellors@mango-solutions.com](mailto:msellors@mango-solutions.com)

Claim processing is human  
powered, labour intensive and  
**slow!**

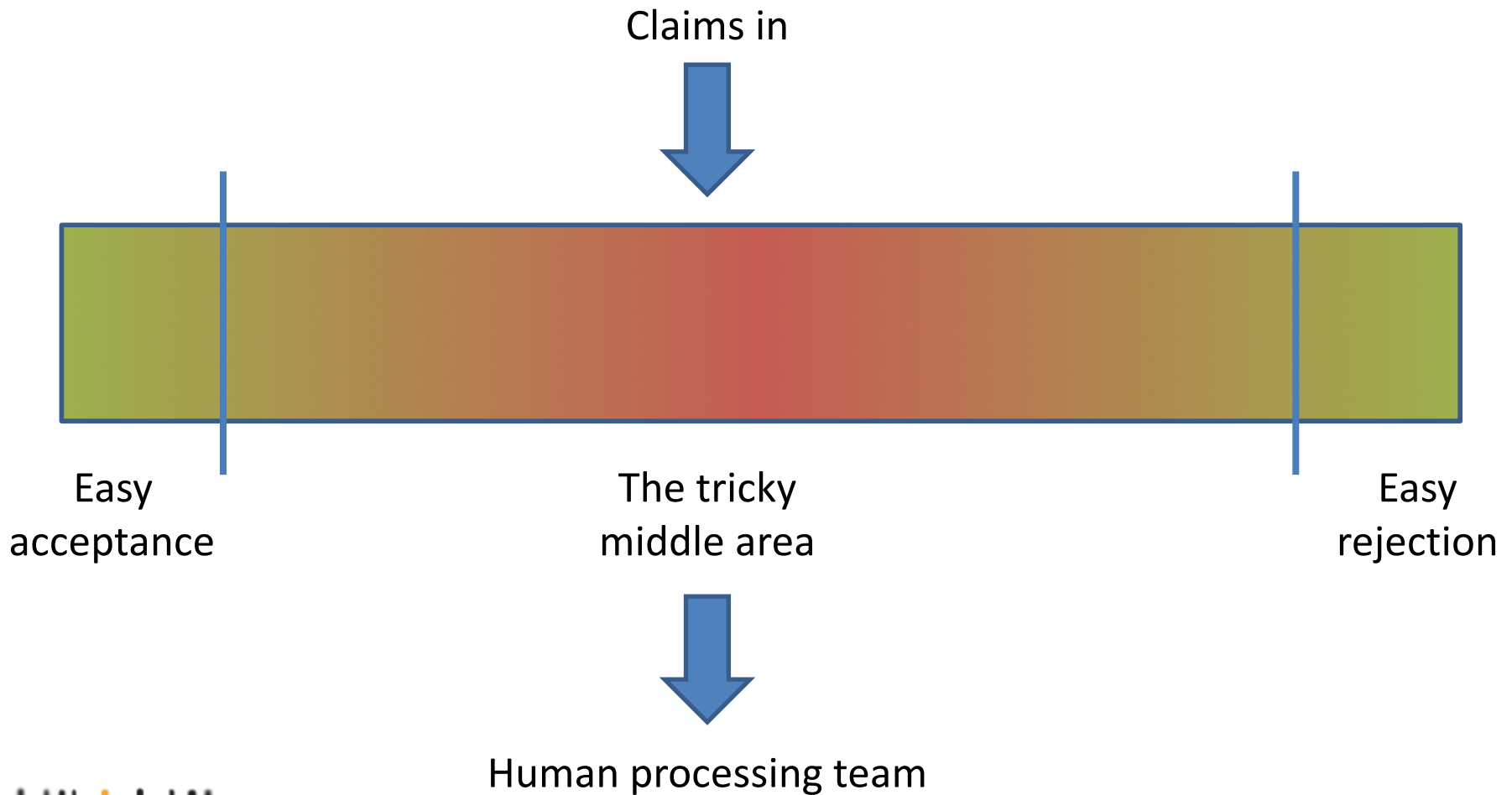




# Old workflow

- Claim received
- Passed to human processing team
- Claim is processed
- Result decided
- Result conveyed to customer

# How can we optimise?



# New Workflow

- Claim received
- Claim form scanned
- OCR performed and converted to XML representation
- XML file is passed to an R service
- Claim is compared against past claims and scored
- Score determines next action:
  - Immediate payout
  - Immediate refusal
  - Passed to processing team for further action

# Workflow

- Claim received
- Claim form scanned
- OCR performed and converted to XML representation
- XML file is passed to an R service
- Claim is compared against past claims and scored
- Score determines next action:
  - Immediate payout
  - Immediate refusal
  - Passed to processing team for further action

Any function you create in R can  
be turned into a service





Website

Inputs

Results

Inputs

Results



# Creating an API in R

- RServe
- OpenCPU
- Plumber
- DeployR
- Probably others

# Other Considerations

- Utilise big data R packages
  - Bigmemory, uses C, stores data in memory
  - ff package, Fast File lookup, not stored in memory
- Build single purpose components where possible
- Ensure that dataset creation is considered as a crucial aspect of design
- Embed into end users workflows, not the other way round

## > **summary(talk)**

- Deploying R as a service (API) is increasingly straightforward
- Quicker, Faster, Stronger
  - Heavily optimised functions and data flows work best
- Put the business challenge before the analytic problem

# Thanks!

