Killing a legacy

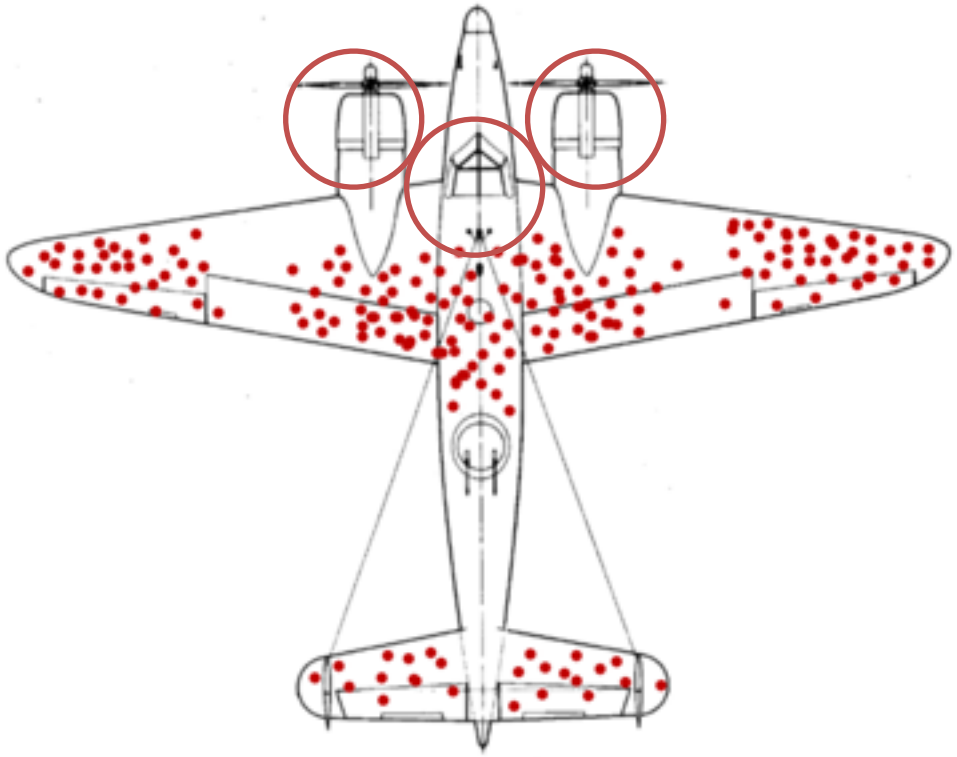
Damien Malone

July 2017

- **Overview – Issues**

- Volume of change
- Attracting the right fit employees
- Employee Motivation
- Aging employee demographic
- Support Business Strategy

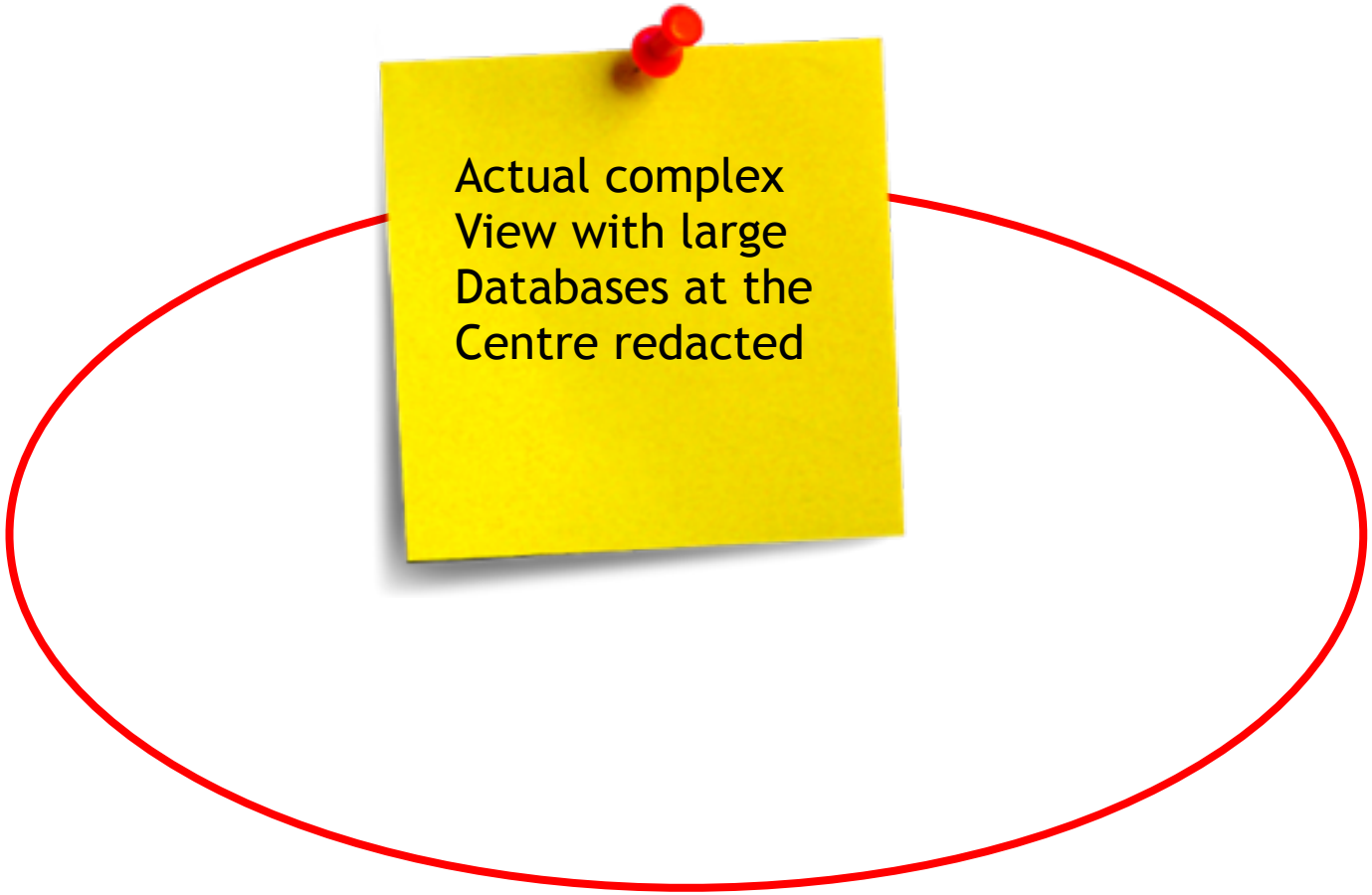
- **Beware of success stories**



- Where I Live

- What I see (What good looks like)

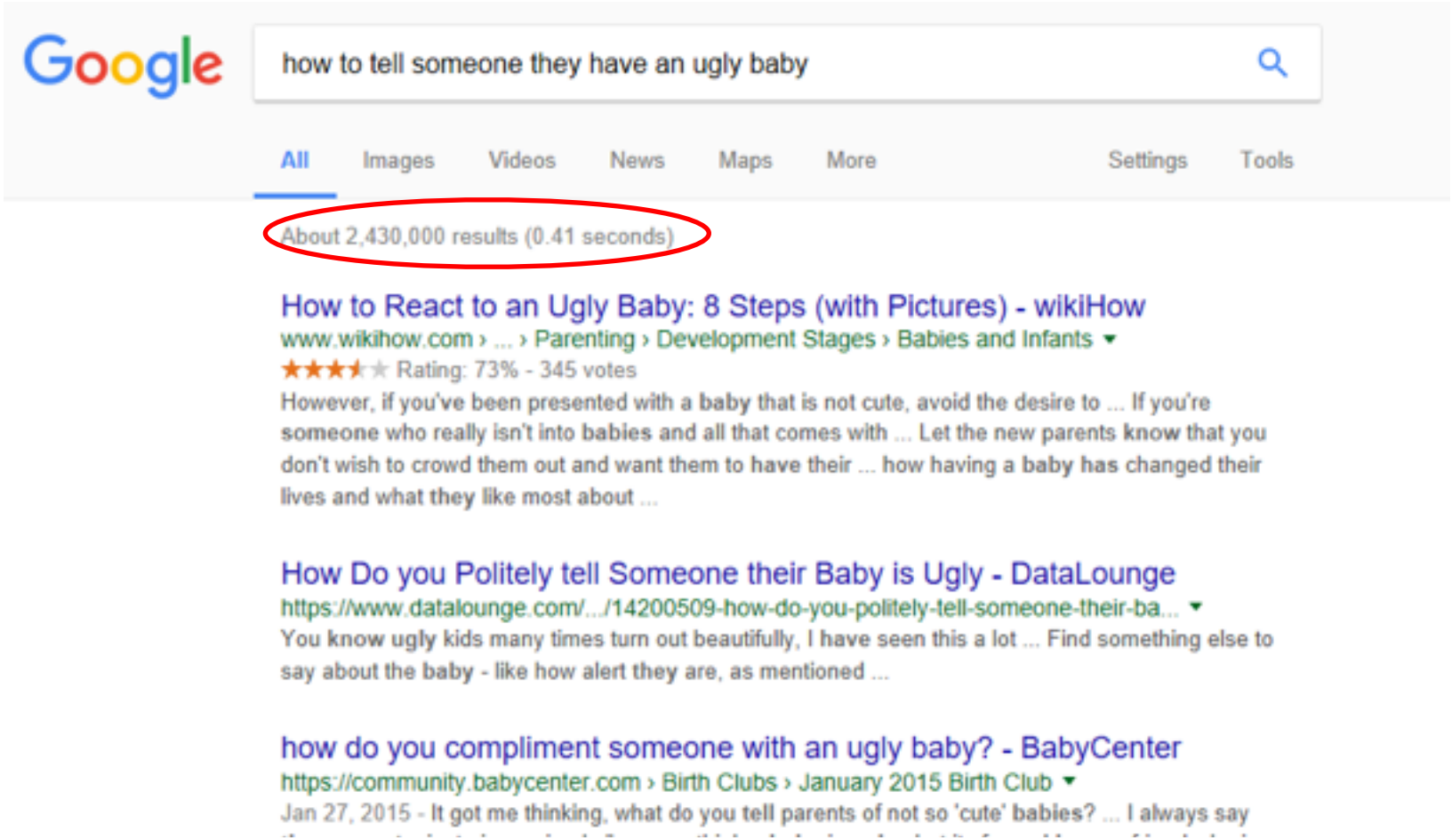
- Whats really there



- **Killing the Legacy**

1. Peoples Legacy
    - People working on the legacy system may be reluctant to help you replace '*their*' system, take their knowledge/power/job
    - The senior management you need to convince to fund this may have built their career on the back of this implementation, its their baby.

- **Peoples Legacy**



- **Killing the Legacy**

2. Legacy Systems
    - 3 Tier 1 systems that were legacy
        - Website
        - Core partner/policy/billing
        - In-house claims system

- **Legacy Systems**

All three systems

10< Years Old <20

We were risk averse in original product selection

Approx. €3b transactions between them

- **Approach 1 – Web Stack**

>100k registered users

Small sales funnel (€10m)

Large lead generator/Information source

Bespoke MVC pattern, ~14years old

- **Approach 1 – Web Stack - Issues**

- Old technologies
- Slow turnaround on change
- Difficulty if getting skilled staff to work on it/support
- Old look and feel with non-responsive front end

- **Approach 1 – Web Stack - Solution**

- Rip and replace easier than remediation
- One foul swoop
- Get business buy-in that no back end system changes are part of the project

- **Approach 1 – Web Stack - Solution**

Pros

No scope creep

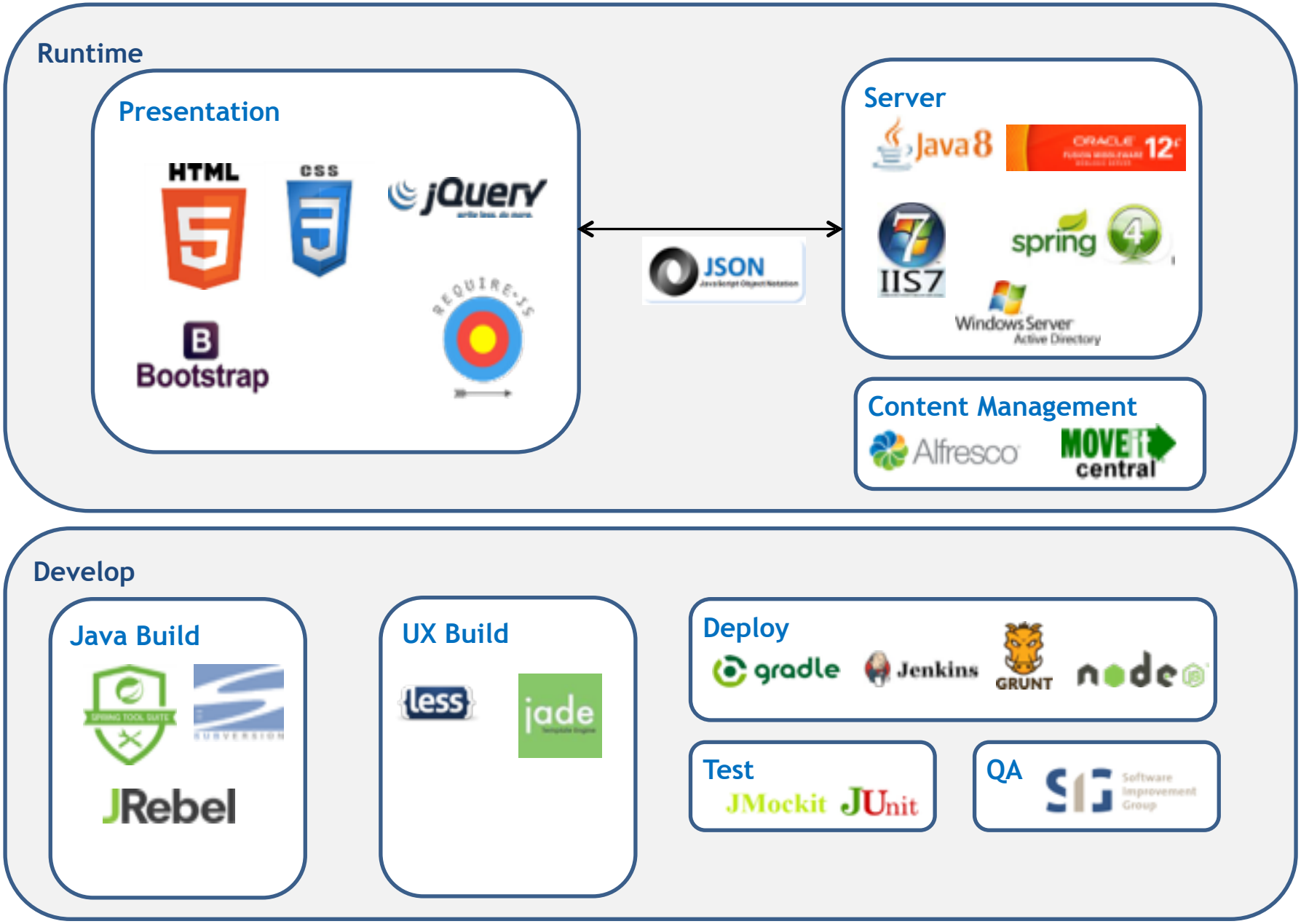
Suited waterfall methodology

Full control of delivery, no dependencies

Cons

No business uplift for 5 months with most of their budget blown

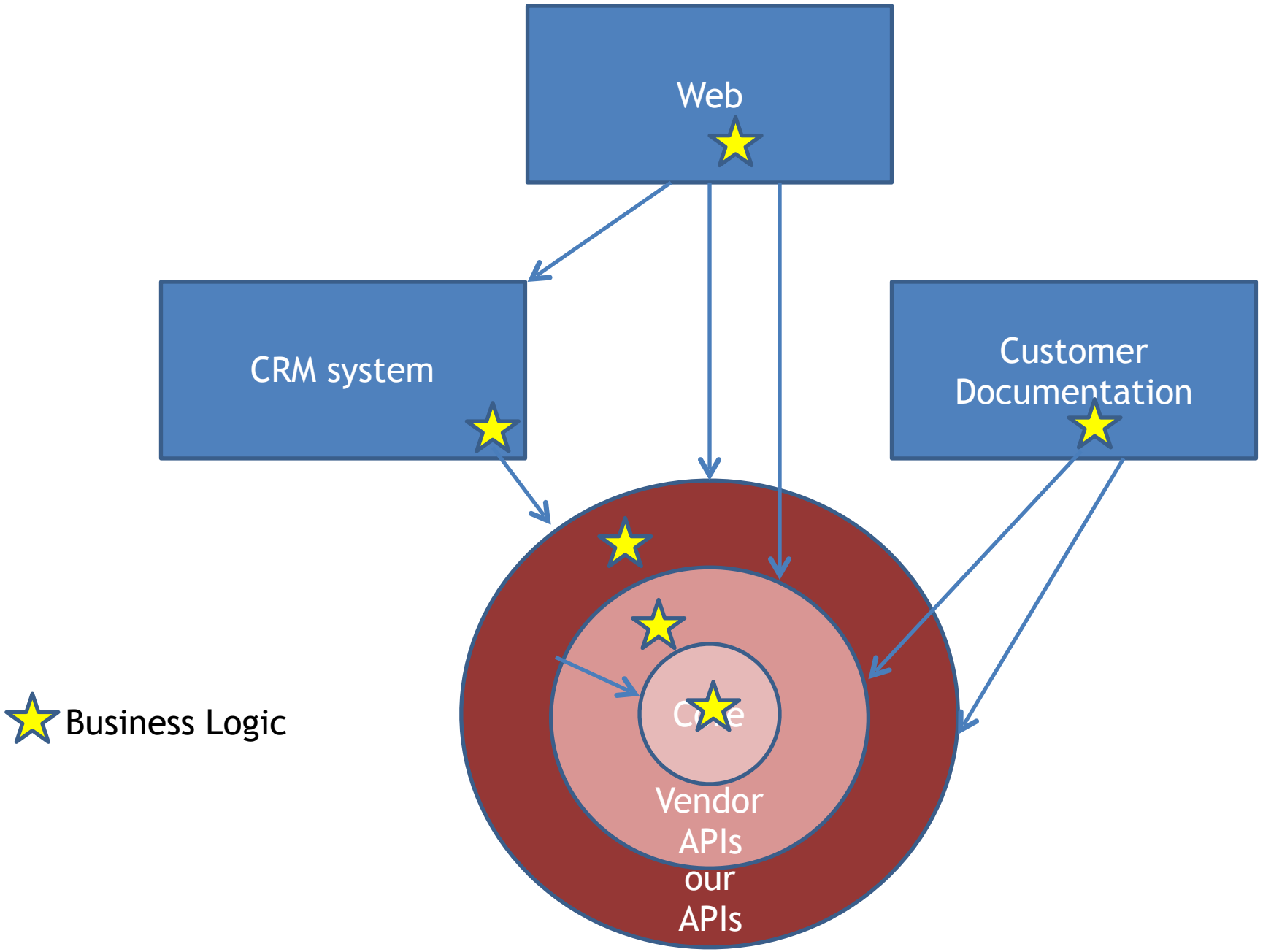
Web Technology Stack



- **Approach 2 –Policy System**

- **Centre of our universe**
- **Oracle forms/DB system**
- **Holds all partner/policy/Billing information**

- **Issues**


Web
CRM system
Customer Documentation
Core
Vendor APIs
our APIs
Business Logic

- # Issues - continued

Have you seen method names like this before???

SavePolicy(….)
SavePolicy1(….)
SavePolicyWeb(….)
SavePolicyLCRProject(….)

Layering without governance is just a more complicated mess

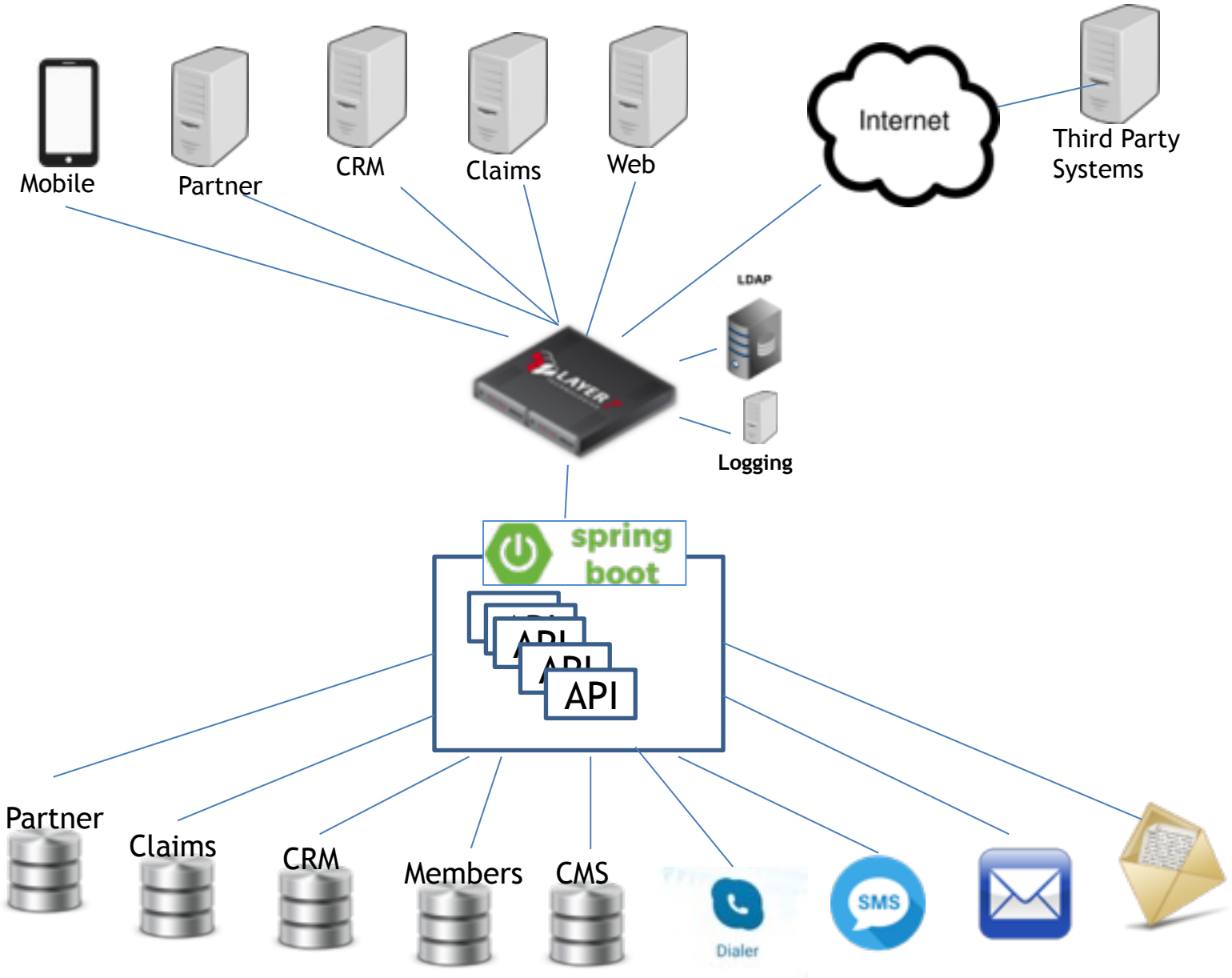
- **Approach 2 –Solution**

- Not the right time to rip and replace
- Remediation too large for a system that is near EOL

- For COTS going forward
  - OOTB
  - Configuration
  - Customisation (Last resort)

- Hide issues, centralise. Place System behind an API gateway

Approach 2 -Solution



• Approach 2 –Solution

Pros

Hides complexities from downstream systems

Enables rip and replace in the future

- Catalogue of dependent systems
- Single point to replace/test

Enables REST api integration

Moves the DB away from the centre of the landscape

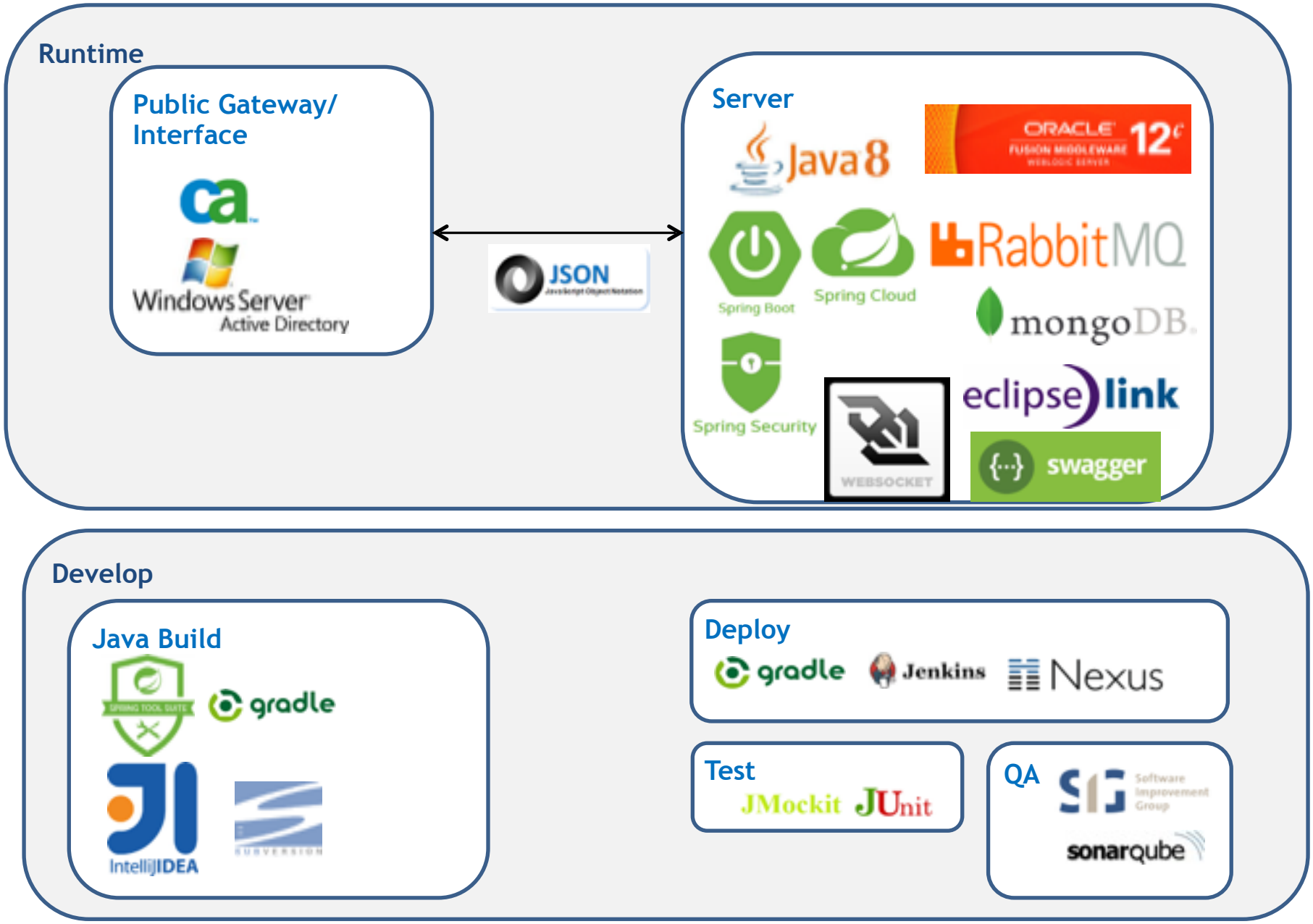
Allows for reusable components

Cons

Legacy System still on landscape

At times of pressure, should we take the old route

Enterprise Integration Stack



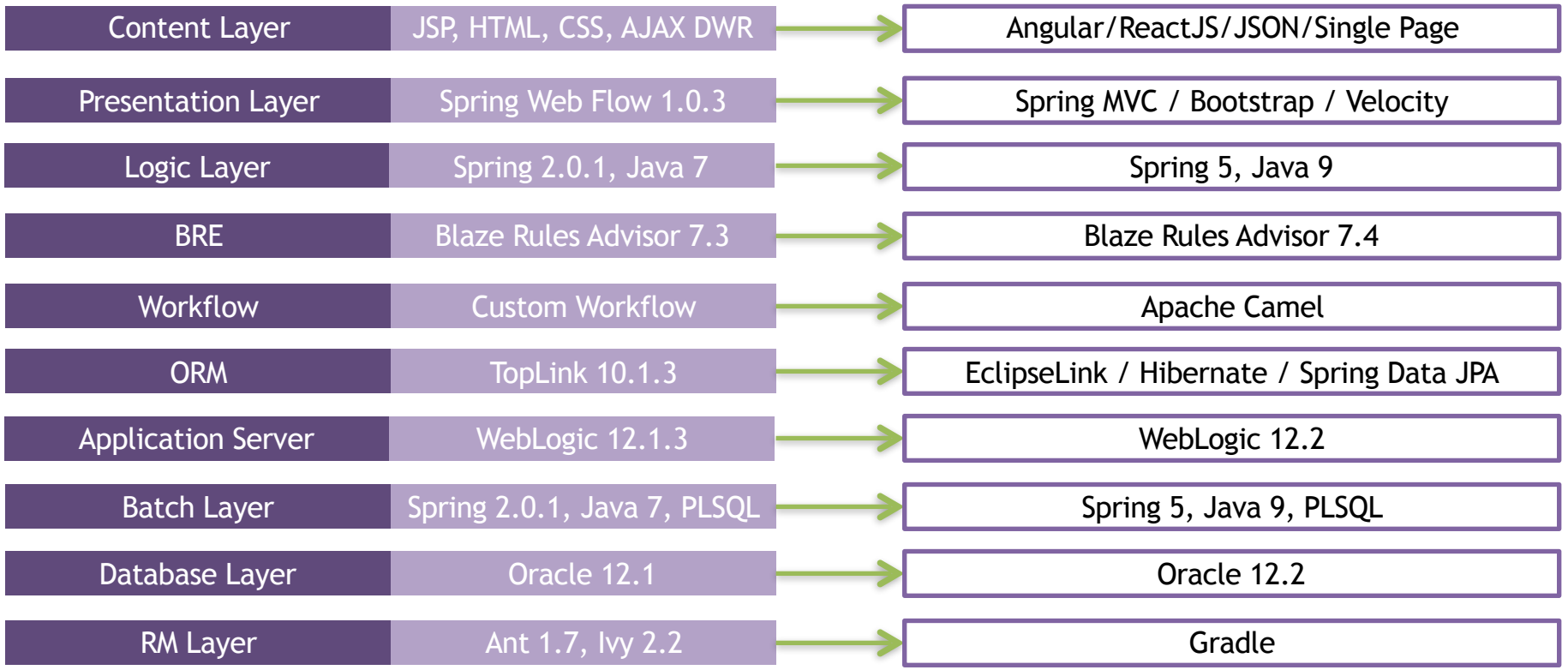
- Approach 3 – Claims System

- Bespoke claims system
- Large man years effort
- Large financial transactions per year
- Has had no technology investment in 10 years
- Risk adverse product selection to begin with

- Approach 3 – Claims issues

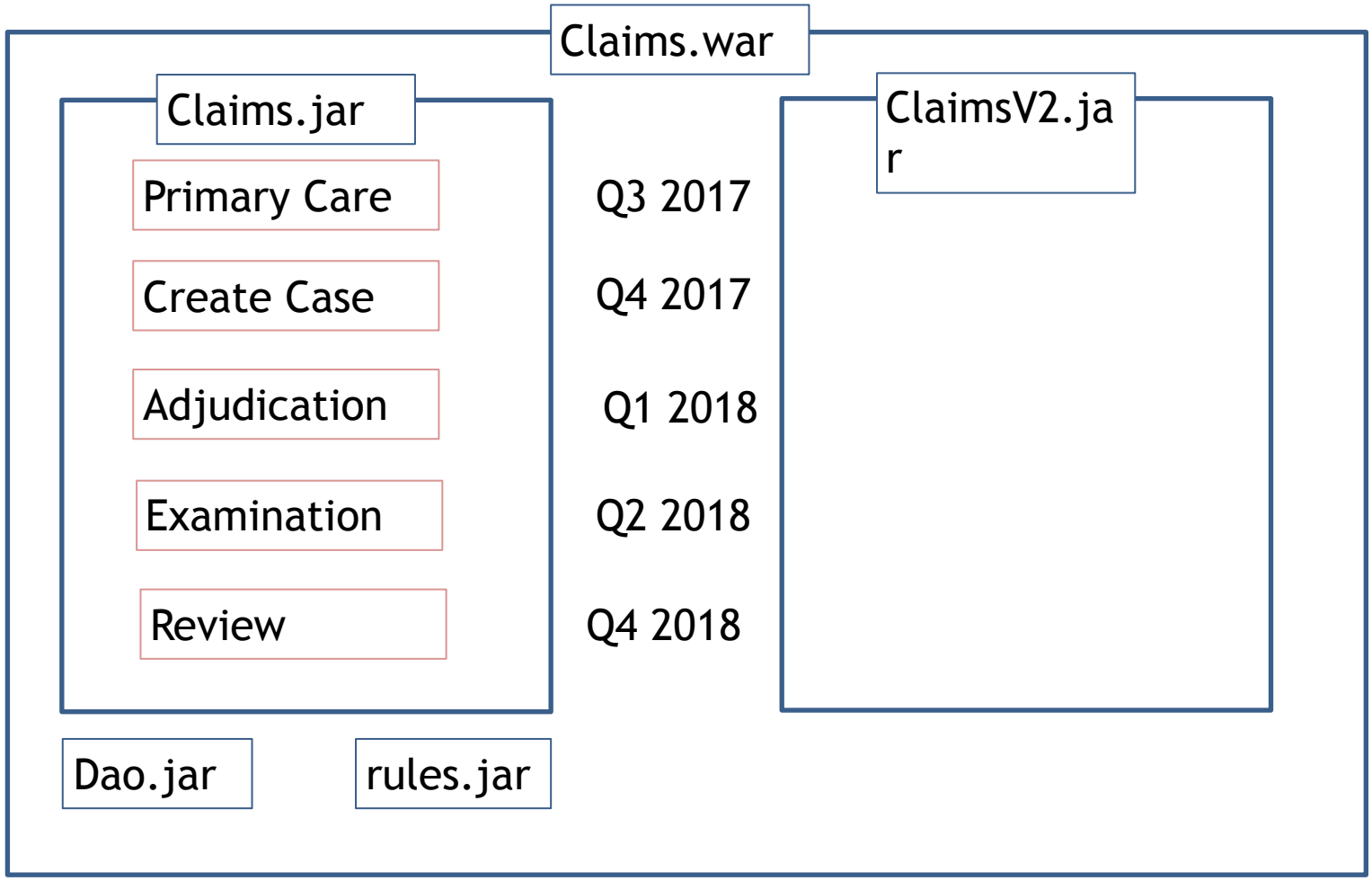
- Unwillingness of developers to work on this
- Outsourcing teams couldn’t be ramped up
- 2 speed IT leading to demotivation
- Technologies running out of support
- Speed of enhancements
- Architecture struggles to support multiple sites(chatty)

- Approach 3 – Solution



- Approach 3 – Parallel Upgrade

- Parallel Upgrade



- Approach 3

- Pros
  - Allows for continued business value delivery
  - Costs can be offset against change projects
  - No big bang

- Cons
  - Compatibility issues
  - Duplication for a time
  - The long goodbye, still need to support legacy for a time

- **Killing Legacies Words of Warning**

Technical Debt

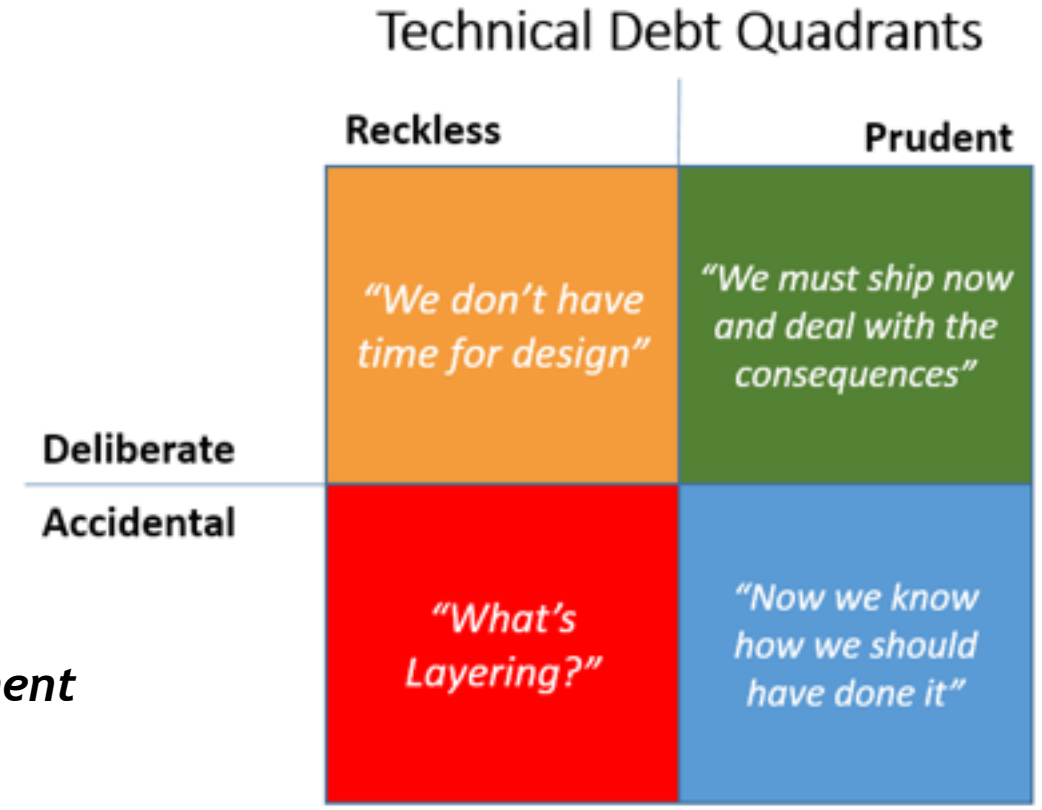
Don’t confuse mindful

TD with Reckless TD

that you have no intention of

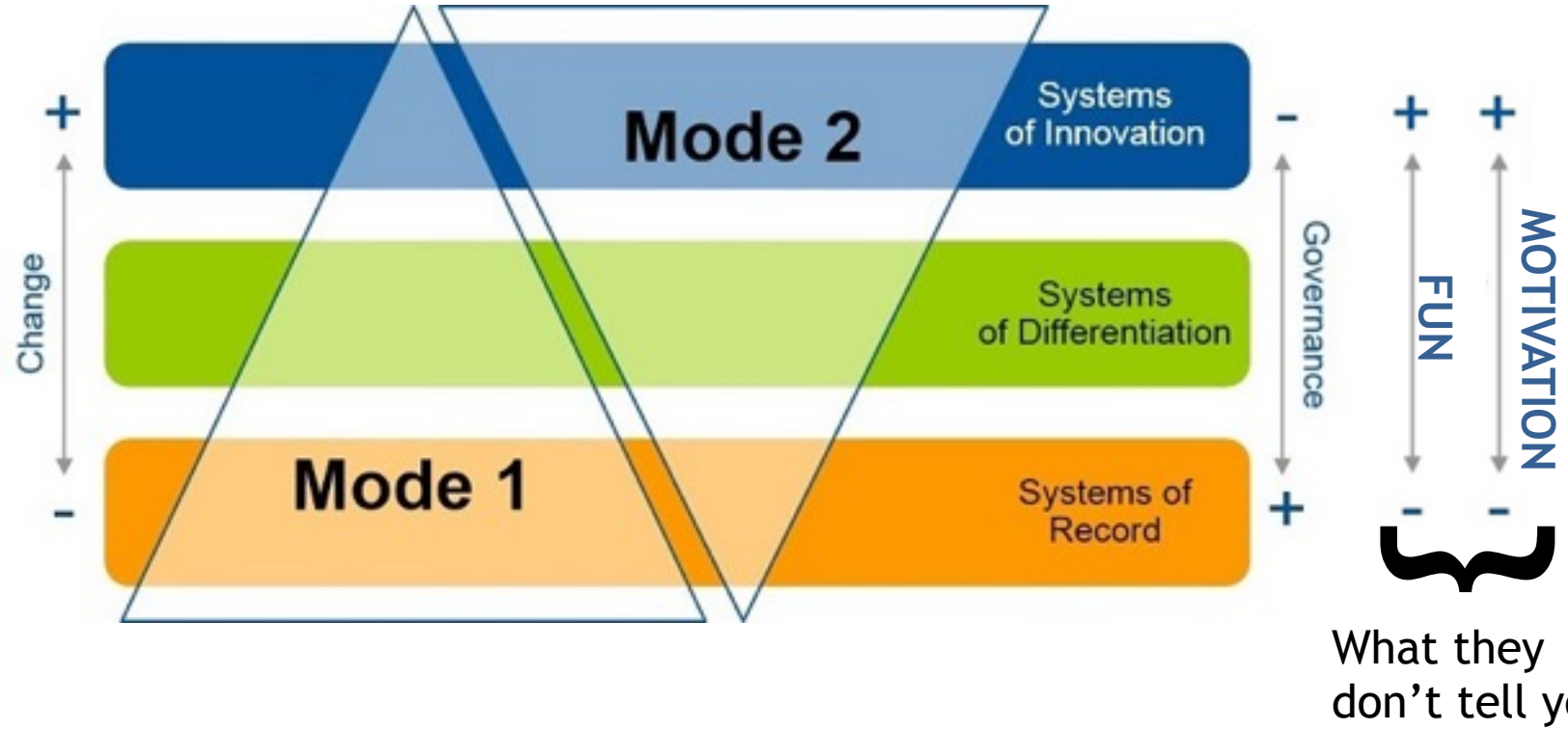
repaying

There is nothing more permanent than a short term fix



- **Killing Legacies Words of Warning**

Bi-modal IT



What they don't tell you

- **Killing Legacies Words of Warning**

Don’t be afraid to move back into the ivory tower (on rare occasions!!)

Just don’t be a Princess about it

Software and Hardware are easy. Peopleware is the hard part.

Q&A