

SPA 2015

28<sup>th</sup> June – 1<sup>st</sup> July 2015

# Last Responsible Moment or Big Ball of Mud?

OUTPUTS



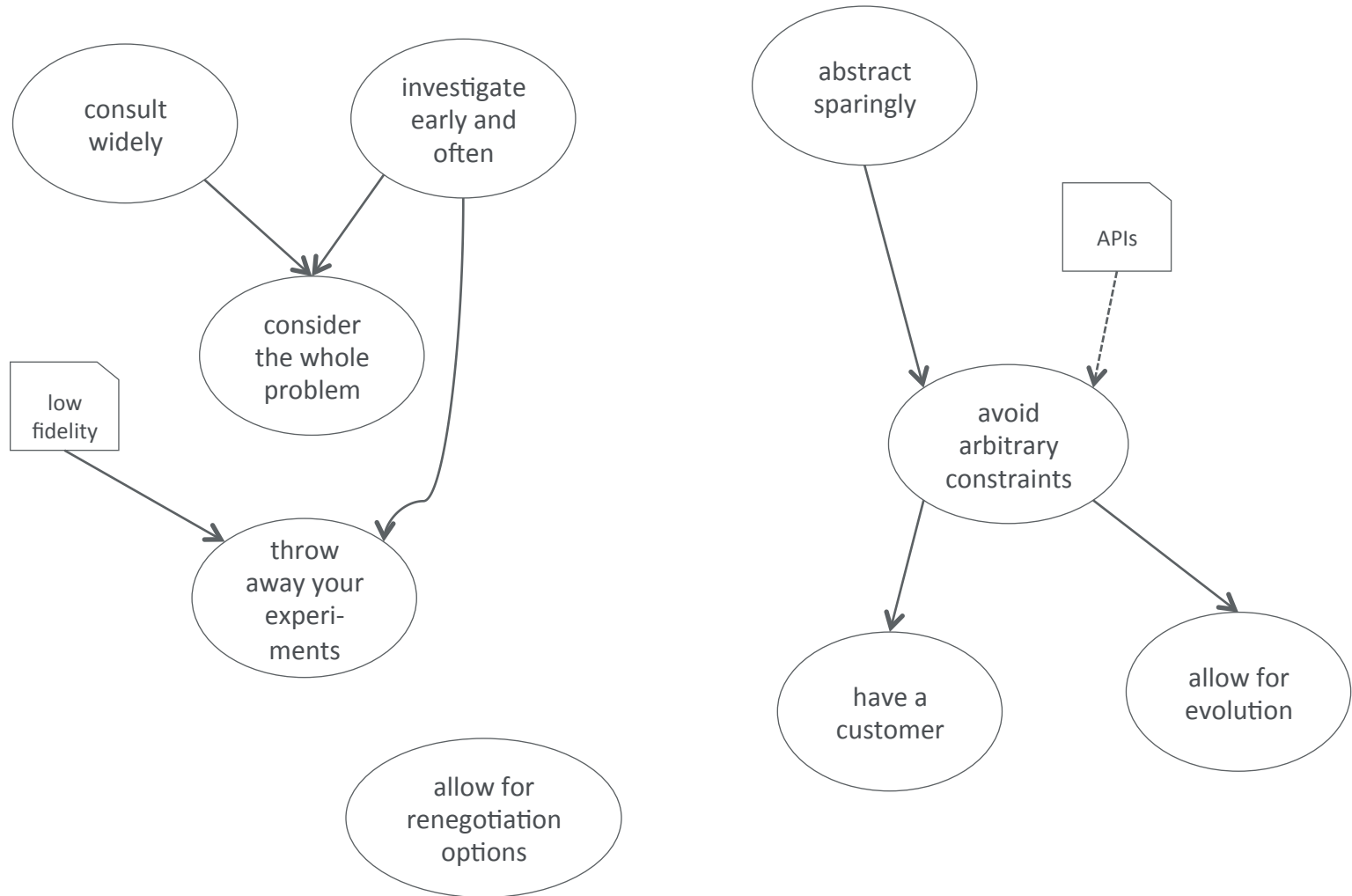
# Topics and Ideas

- Fear of making the wrong decision – allow people to fail
- Decision scope – how much is affected by a decision
  - smaller scope => impact of a wrong decision is less and the risk of deferral is also less
- Decision longevity (how long the impact is felt for)
  - shorter longevity => impact of a wrong decision is less and the risk of deferral is also less
- “Real options” – an interesting idea but the practicality of assigning sound metrics is debatable
- Software pace of change (eg new versions of software products)
  - will today’s decision still be the right one in the future?
- Risk appetite (a higher risk appetite would allow for decisions to be deferred more)
- Service or API design – where should this be on the deferral scale? depends on stakeholders’ knowledge of the domain
- The ‘S’ curve of innovation over time, identified in 1900’s by Gabriel Tarde
  - paper here on later work:  
[http://web.mit.edu/mamd/www/tech\\_strat/courseMaterial/topics/topic3/readings/Exploring\\_the\\_Limits\\_of\\_the\\_Technology\\_S-Curve/Exploring\\_the\\_Limits\\_of\\_the\\_Technology\\_S-Curve.pdf/Exploring\\_the\\_Limits\\_of\\_the\\_Technology\\_S-Curve.pdf](http://web.mit.edu/mamd/www/tech_strat/courseMaterial/topics/topic3/readings/Exploring_the_Limits_of_the_Technology_S-Curve/Exploring_the_Limits_of_the_Technology_S-Curve.pdf/Exploring_the_Limits_of_the_Technology_S-Curve.pdf)

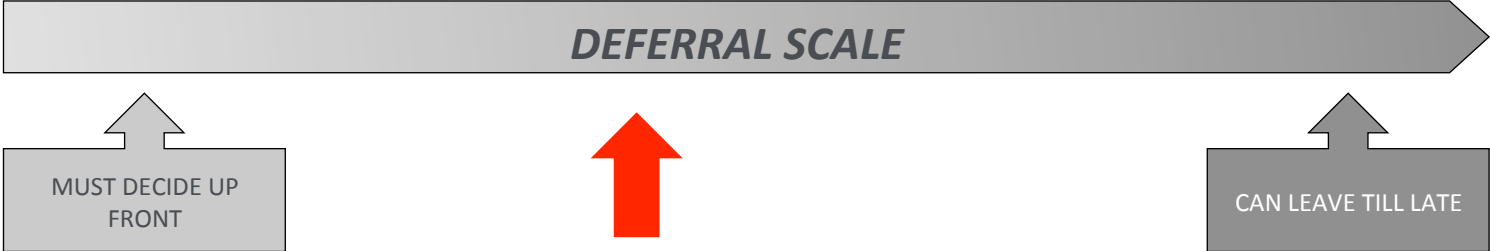
# Deferral Strategies: Group 1

- Buy options – eg training
- Move functionality away from the variation point – eg thinner UI
- Low-fi first
  - technical debt concept
  - gather more data
- Letter of the law – appear to use but not actually / core
- Transparent cost
  - buy political space to defer
  - initial cost + carrying cost
- Set-based engineering – build two, choose one

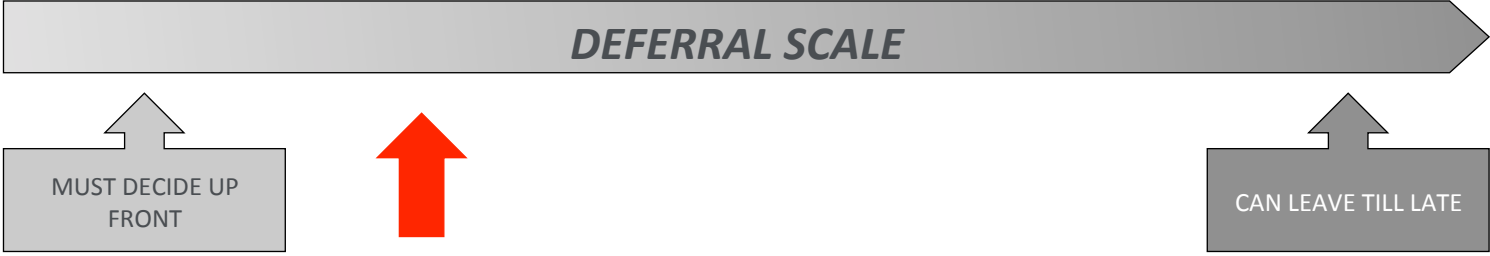
# Deferral Strategies: Group 2




# Decision Proforma

<b>Context</b>	
<b>Decision</b>	Search technology used
<b>Notes</b>	requirements risk mitigation
<b>Position on Deferral Scale</b>	 <p>The diagram shows a horizontal grey arrow pointing right, labeled <i>DEFERRAL SCALE</i>. Below the arrow, there are three boxes. On the left, a grey box labeled "MUST DECIDE UP FRONT" has a grey arrow pointing up to the start of the scale. In the center, a red arrow points up to the middle of the scale. On the right, a grey box labeled "CAN LEAVE TILL LATE" has a grey arrow pointing up to the end of the scale.</p>
<b>Deferral Tactics</b>	<ul style="list-style-type: none"><li>• decouple storage (somehow)</li></ul>

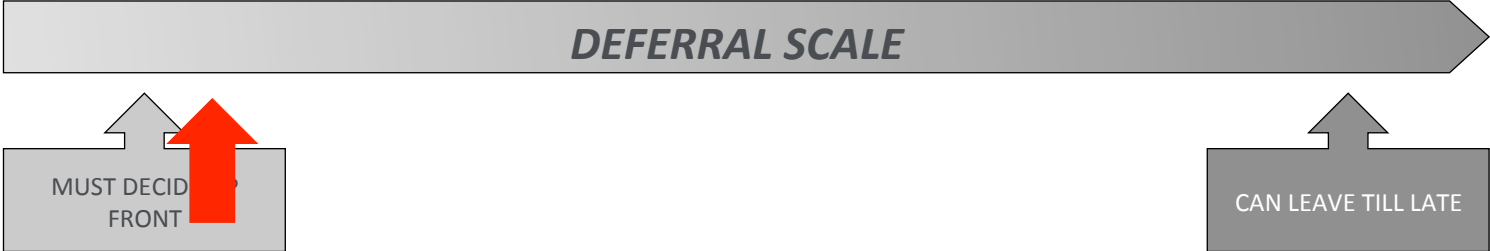
# Decision Proforma

<b>Context</b>	
<b>Decision</b>	Choosing a cloud platform
<b>Notes</b>	How hard is it to switch cloud providers Lowest common denominator
<b>Position on Deferral Scale</b>	 <p>The diagram shows a horizontal grey arrow pointing right, labeled <i>DEFERRAL SCALE</i>. Below the arrow, there are three elements: a grey box on the left containing the text "MUST DECIDE UP FRONT" with a grey arrow pointing up to the start of the scale; a red arrow pointing up to the middle of the scale; and a grey box on the right containing the text "CAN LEAVE TILL LATE" with a grey arrow pointing up to the end of the scale.</p>
<b>Deferral Tactics</b>	

# Decision Proforma

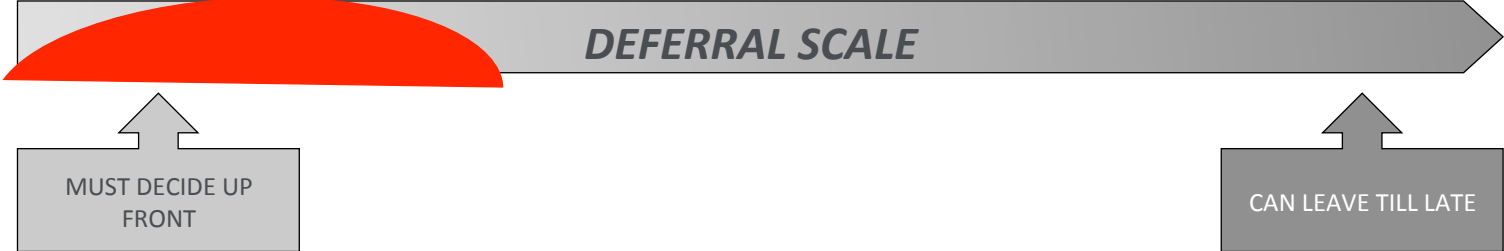
<b>Context</b>	DB technology – ORM? not database devs
<b>Decision</b>	ORM or hand-roll? which ORM?
<b>Notes</b>	
<b>Position on Deferral Scale</b>	 <p>The diagram shows a horizontal grey arrow pointing right labeled "DEFERRAL SCALE". Below the arrow, there are three boxes: "MUST DECIDE UP FRONT" on the left, "CAN LEAVE TILL LATE" on the right, and a red arrow pointing up in the center. Hand-drawn scribbles are present between the red arrow and the "CAN LEAVE TILL LATE" box.</p>
<b>Deferral Tactics</b>	<ul style="list-style-type: none"><li>• Indirection layer – “repository”</li><li>• ‘Buy an option’ – use ORM in a way that could be implemented by plain SQL more easily</li><li>• get training in ORM and SQL</li></ul>

# Decision Proforma

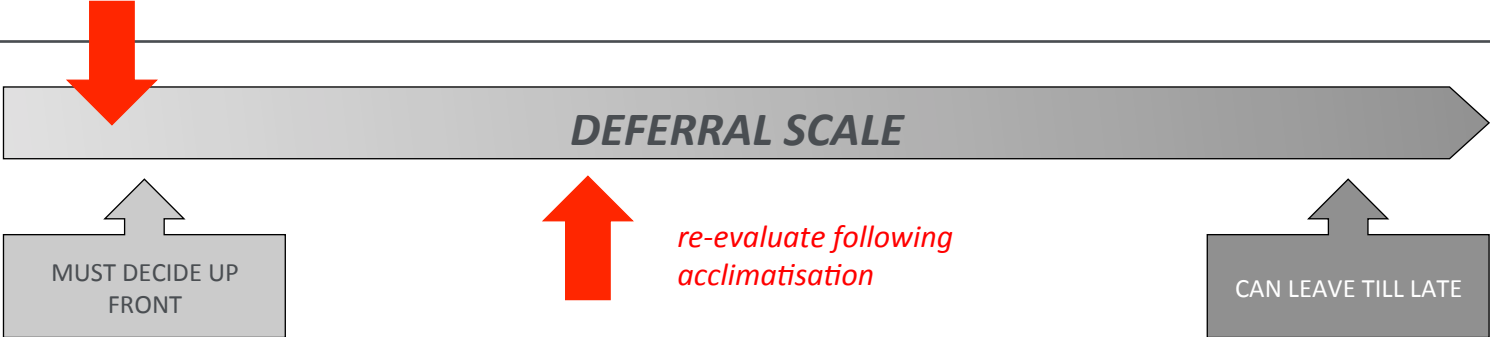
<b>Context</b>	App building in startup – Android? iOS?
<b>Decision</b>	Android first? iOS first? Both in parallel?
<b>Notes</b>	Sunk learning; market share; front-end vs back-end; cross-platform? Given you are going to start with one, when do you start the 2nd
<b>Position on Deferral Scale</b>	 <p>The diagram shows a horizontal arrow pointing right labeled "DEFERRAL SCALE". Below the left end is a box labeled "MUST DECID FRONT" with a grey arrow pointing up and a red arrow pointing up. Below the right end is a box labeled "CAN LEAVE TILL LATE" with a grey arrow pointing up.</p>
<b>Deferral Tactics</b>	<ul style="list-style-type: none"> <li>• push as much into the cloud / backend – make frontend as thin as possible (move functionality away from the variation point)</li> <li>• lower-fidelity version or focus on features that can be cross-platform (eg HTML5/JS web app) features that need native access later (Ward’s technical debt)</li> <li>• analyse market (while doing the previous two)             <ul style="list-style-type: none"> <li>• gather more information to inform the decision</li> <li>• work out how to make progress while gathering information or to help gather information</li> </ul> </li> </ul>



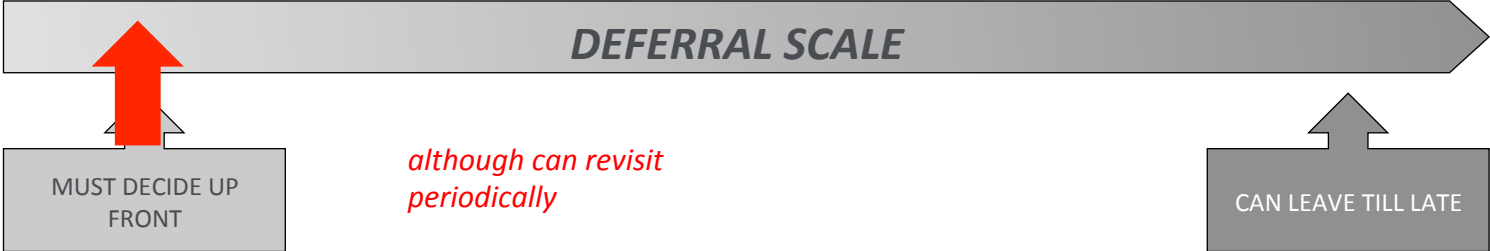
# Decision Proforma

<b>Context</b>	Web-based forms submission app
<b>Decision</b>	Use a service bus?
<b>Notes</b>	junk cost; political pressure; skills and knowledge => training
<b>Position on Deferral Scale</b>	
<b>Deferral Tactics</b>	<ul style="list-style-type: none"> <li>• abstraction layer</li> <li>• “letter vs spirit” conformance</li> <li>• make costs transparent             <ul style="list-style-type: none"> <li>• training costs</li> <li>• carrying costs of additional transparency</li> </ul> </li> </ul>

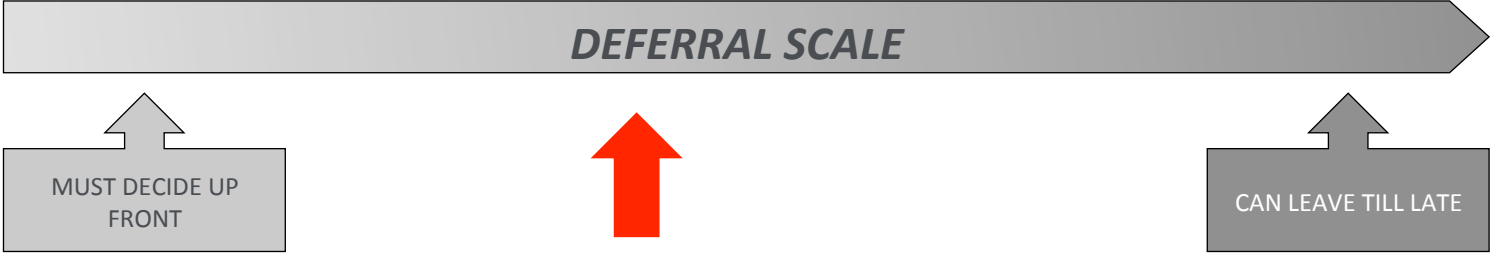
# Decision Proforma

<b>Context</b>	New services slowly strangling a monolith
<b>Decision</b>	Which database to use for multi-region, multi-master systems
<b>Notes</b>	services need to be delivered quicker than ideal evaluation period
<b>Position on Deferral Scale</b>	 <p>The diagram shows a horizontal grey arrow pointing right, labeled "DEFERRAL SCALE". A large red arrow points down to the start of the scale. Below the scale, there are three boxes: "MUST DECIDE UP FRONT" with a grey arrow pointing up to the start of the scale; "re-evaluate following acclimatisation" with a red arrow pointing up to the middle of the scale; and "CAN LEAVE TILL LATE" with a grey arrow pointing up to the end of the scale.</p>
<b>Deferral Tactics</b>	<ul style="list-style-type: none"> <li>• variation point</li> <li>• explore options in parallel</li> <li>• provisional decisions – criteria that need to hold over time</li> <li>• question: who requires decisions? why do they require it now?</li> </ul>


# Decision Proforma

<b>Context</b>	Product transition approach (from legacy to new technology)
<b>Decision</b>	Use code conversion and emulation techniques (usually 3 <sup>rd</sup> -party framework)
<b>Notes</b>	20 million lines of Progress ABL code, moving to Java other options: rewrite; application strangulation
<b>Position on Deferral Scale</b>	 <p>The diagram shows a horizontal arrow labeled "DEFERRAL SCALE" pointing to the right. Below the left end of the arrow is a box labeled "MUST DECIDE UP FRONT" with a red arrow pointing up to the arrow's tail. Below the right end of the arrow is a box labeled "CAN LEAVE TILL LATE" with a grey arrow pointing up to the arrow's tip. In the center of the scale, the text "<i>although can revisit periodically</i>" is written in red.</p>
<b>Deferral Tactics</b>	<ul style="list-style-type: none"> <li>• Prototype</li> <li>• Renegotiation points: agreement that we can change decision, given new information (criteria for decision change)</li> <li>• Bring system under test at the API (trade entry point to the system): deferral? or just doing something else?</li> </ul>

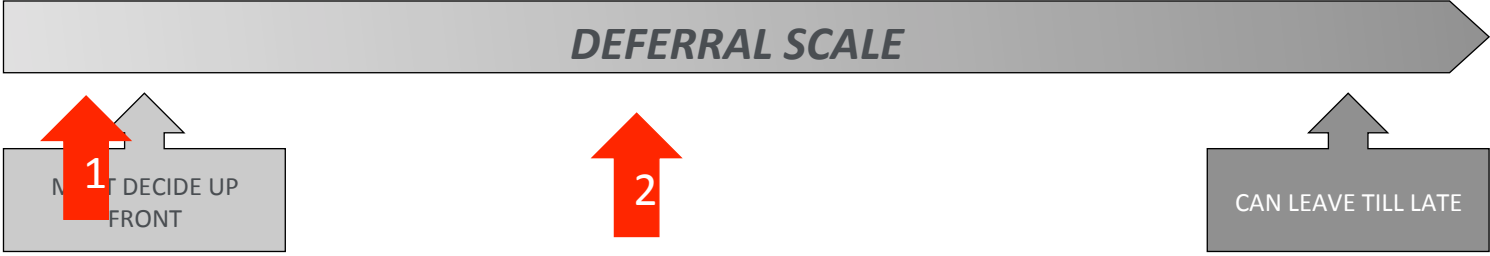
# Decision Proforma

<b>Context</b>	Use of Chef for deployment automation
<b>Decision</b>	Choose approach to infrastructure config management
<b>Notes</b>	Other options include: doing it manually, other scripting languages, Puppet / Cf Engine / Salt
<b>Position on Deferral Scale</b>	 <p>The diagram shows a horizontal grey arrow pointing right, labeled "DEFERRAL SCALE". Below the arrow, there are three boxes. The left box is labeled "MUST DECIDE UP FRONT" and has a grey arrow pointing up to the start of the scale. The middle box is a red arrow pointing up to the center of the scale. The right box is labeled "CAN LEAVE TILL LATE" and has a grey arrow pointing up to the end of the scale.</p>
<b>Deferral Tactics</b>	<ul style="list-style-type: none"><li>• A   B testing</li><li>• Prototypes</li></ul>


# Decision Proforma

<b>Context</b>	Extracting service from monolith to aid switching DB technology
<b>Decision</b>	What DB technology should be switched to?
<b>Notes</b>	Service is extracted, DB is the same. New tech discussed but very little investment. Waiting for additional research before final decision. Ongoing cost of maintainin current DB tech
<b>Position on Deferral Scale</b>	
<b>Deferral Tactics</b>	<ul style="list-style-type: none"> <li>• Further investment in existing tech while research takes place =&gt; risk of having to make snap decision if limits of current tech reached</li> <li>• cost of not making decision – can it be brought forward?             <ul style="list-style-type: none"> <li>• lean on research?</li> <li>• reduces uncertainty in cost of changing tech</li> </ul> </li> </ul>

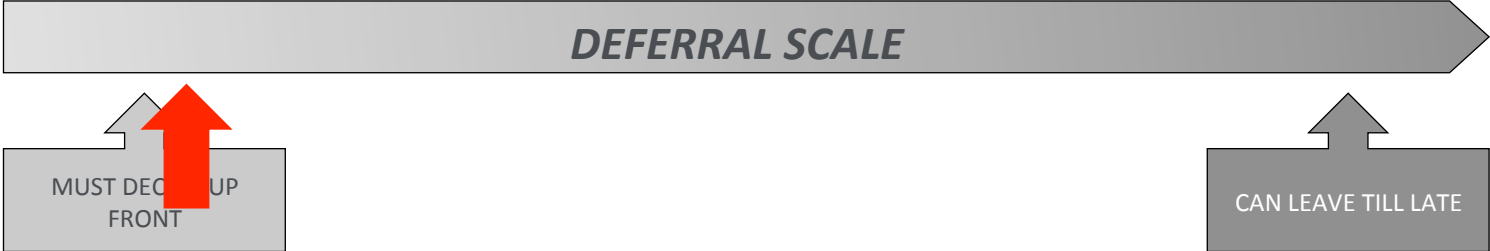
# Decision Proforma

<b>Context</b>	Investing in a home-built high-performance log aggregation system
<b>Decision</b>	1) prototype 2) further investment
<b>Notes</b>	limit cost by only committing to small investment
<b>Position on Deferral Scale</b>	 <p>The diagram shows a horizontal arrow labeled "DEFERRAL SCALE". Below the arrow, there are three boxes. The left box contains "DON'T DECIDE UP FRONT" with a red arrow "1" pointing up to it. The middle box contains "2" with a red arrow "2" pointing up to it. The right box contains "CAN LEAVE TILL LATE" with a grey arrow pointing up to it.</p>
<b>Deferral Tactics</b>	<ul style="list-style-type: none"> <li>• Prototyping</li> <li>• Façade – but lock-in, need for specific features can hurt</li> <li>• Provisional decision – explicit criteria for revisiting</li> <li>• Only commit the bits that matter</li> </ul>

# Decision Proforma

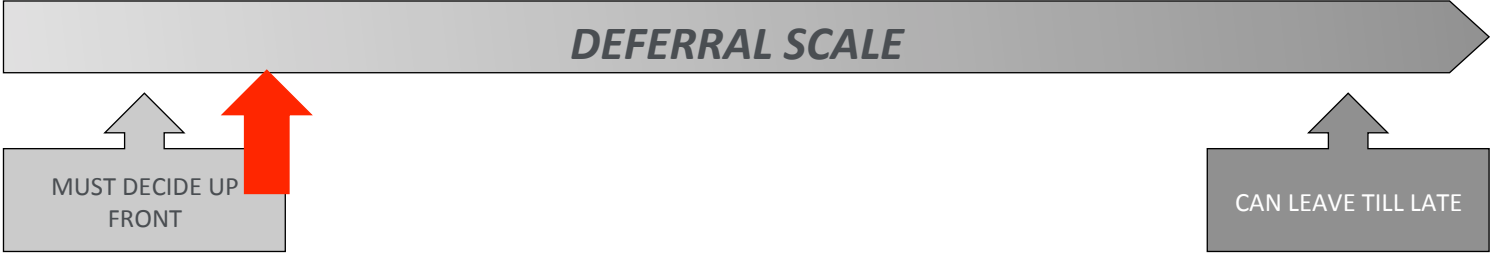
<b>Context</b>	PoC food hygiene mobile inspection system
<b>Decision</b>	NodeJS vs DropWizard for services
<b>Notes</b>	Constraint of 5 sprints (10 weeks)
<b>Position on Deferral Scale</b>	 <p>The diagram shows a horizontal grey arrow pointing right, labeled "DEFERRAL SCALE". Below the arrow, there are three elements: a grey box on the left labeled "MUST DECIDE UP FRONT" with a grey arrow pointing up to the start of the scale; a red arrow pointing up to the middle of the scale; and a grey box on the right labeled "CAN LEAVE TILL LATE" with a grey arrow pointing up to the end of the scale. To the right of the red arrow, there are two bullet points: "• contract constraint" and "• development constraint".</p>
<b>Deferral Tactics</b>	<ul style="list-style-type: none"> <li>• No scope – contract requirement</li> <li>• Renegotiate?</li> </ul>

# Decision Proforma

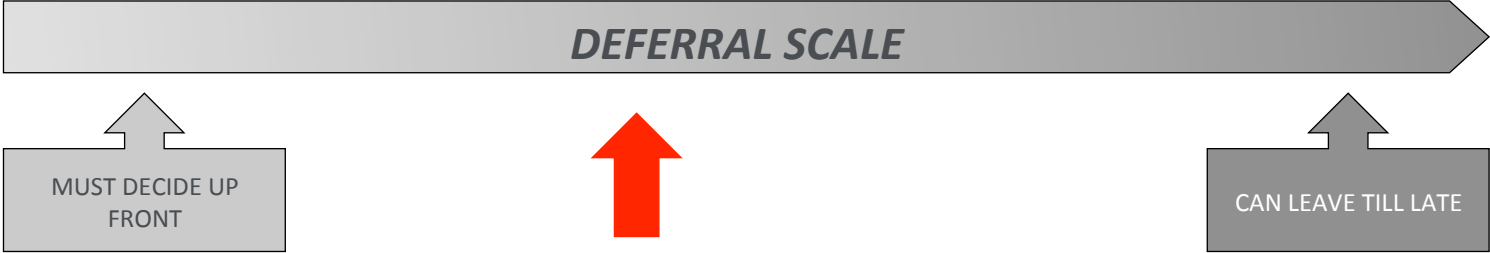
<b>Context</b>	Whether to track state of adverts using Event Sourcing
<b>Decision</b>	Yes! We love event sourcing
<b>Notes</b>	Generalised mechanism to allow the adverts to be used in the future. “Keeps options open”
<b>Position on Deferral Scale</b>	
<b>Deferral Tactics</b>	<ul style="list-style-type: none"><li>• Build API to access historic data (as opposed to all client creating their own ???)</li><li>• Donside –Building API is “upfronting” (too early) and constrain clients</li></ul>



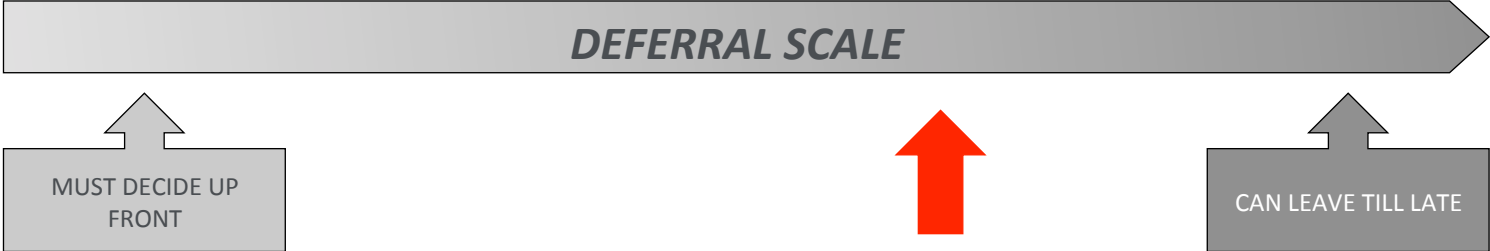
# Decision Proforma

<b>Context</b>	Implementation language for payment processing product to be resold
<b>Decision</b>	Erlang for messaging
<b>Notes</b>	Early decision without understanding of long tail costs and constraints => early decision needing long term view
<b>Position on Deferral Scale</b>	 <p>The diagram shows a horizontal arrow labeled "DEFERRAL SCALE" pointing to the right. Below the left end of the arrow is a box labeled "MUST DECIDE UP FRONT" with a grey arrow pointing up to the start of the scale. A red arrow points up from the left side of the scale. Below the right end of the arrow is a box labeled "CAN LEAVE TILL LATE" with a grey arrow pointing up to the end of the scale.</p>
<b>Deferral Tactics</b>	<ul style="list-style-type: none"><li>• Build a throwaway prototype of the replacement msg switch to gain early insights into impact of this decision</li><li>• There may be stakeholders affected by the decision that you didn't know about. They may reveal important information</li></ul>

# Decision Proforma

<b>Context</b>	Technology to implement and event store on AWS
<b>Decision</b>	DynamoDB
<b>Notes</b>	<p>Step 1: coupled to Dynamo</p> <p>Step 2: generalise the payload</p> <p>Risk: clients need to change ‘technical’ binding. In 3 years Dynamo becomes unaffordable</p>
<b>Position on Deferral Scale</b>	 <p>The diagram shows a horizontal grey arrow pointing right, labeled <i>DEFERRAL SCALE</i>. Below the arrow, there are three boxes: a grey box on the left labeled 'MUST DECIDE UP FRONT' with a grey arrow pointing up to the start of the scale; a red arrow pointing up to the middle of the scale; and a grey box on the right labeled 'CAN LEAVE TILL LATE' with a grey arrow pointing up to the end of the scale.</p>
<b>Deferral Tactics</b>	<ul style="list-style-type: none"> <li>• Mitigation: consider possible future scenarios; consider different dimensions of the problem</li> <li>• To allow deferral: do enough investigation to understand the problem and options</li> </ul>

# Decision Proforma

<b>Context</b>	Choosing (FIX) message processor
<b>Decision</b>	Self build, open source, commercial appliance (CCTS) – open source
<b>Notes</b>	RRM appeared quite suddenly (first one was missed)
<b>Position on Deferral Scale</b>	 <p>The diagram shows a horizontal grey arrow pointing right, labeled "DEFERRAL SCALE". Below the arrow, there are three elements: a grey box on the left containing "MUST DECIDE UP FRONT" with a grey arrow pointing up to the start of the scale; a red arrow pointing up to the middle of the scale; and a grey box on the right containing "CAN LEAVE TILL LATE" with a grey arrow pointing up to the end of the scale.</p>
<b>Deferral Tactics</b>	