CIFER

Community Identity Framework for Education and Research

Rob Carter, Duke University
Keith Hazelton, UW-Madison, Internet2

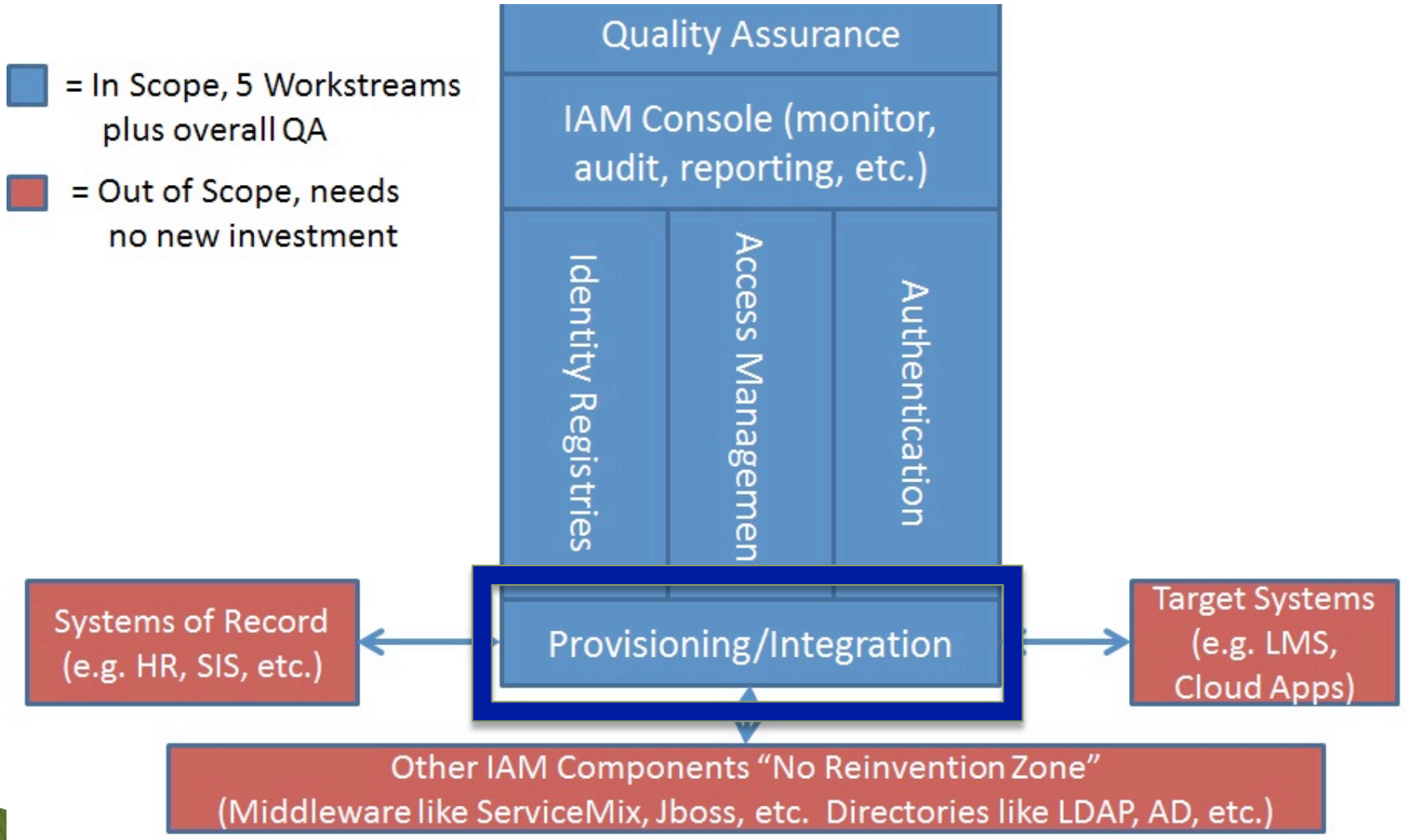
June 10-15, 2012

Growing Community;
Growing Possibilities

What is CIFER, really?

- An alternative business model for higher education IAM
- A developing practice of coordination across existing projects in Kuali, Internet2, Jasig and elsewhere
- A sponsor and coordinator of new development—but only where it has to be
- *NOT* YAOSO

CIFER Workstreams and Scope: P&I



CIFER Provisioning and Integration Timeline

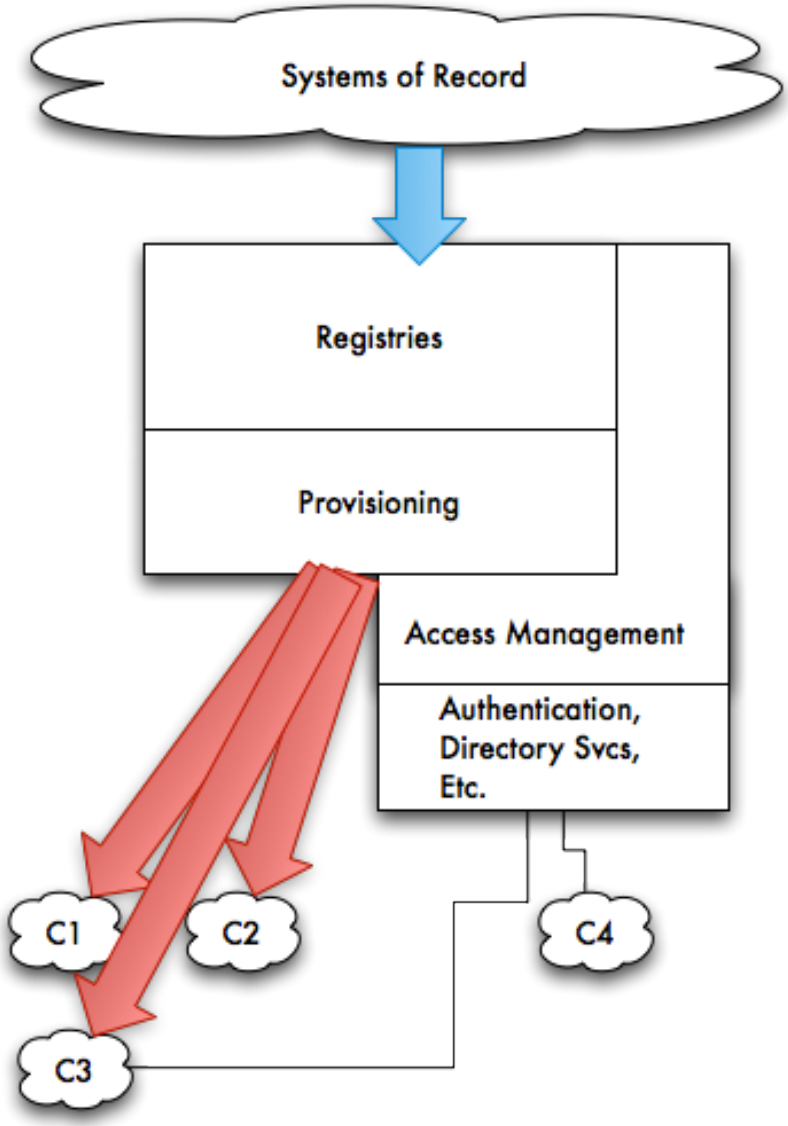
Workstream	Year 1	Year 2
P&I Caretakers: Internet2 (Grouper PSP), Kuali (KIM), Possibly leveraging ForgeRock OpenIDM	• System of Record (SoR) to Registry & Registry to Consumer Toolkits • Connectors to select Consumer systems (email, LMS, library) • Community-contributed System of Record (SoR) to Registry connectors	• Dev. tool plug-ins to accelerate integration • More SoR and Consumer connectors • Business rules & engine for automated ID & affiliation life-cycle management

CIFER P&I

- People Integrating People...Data

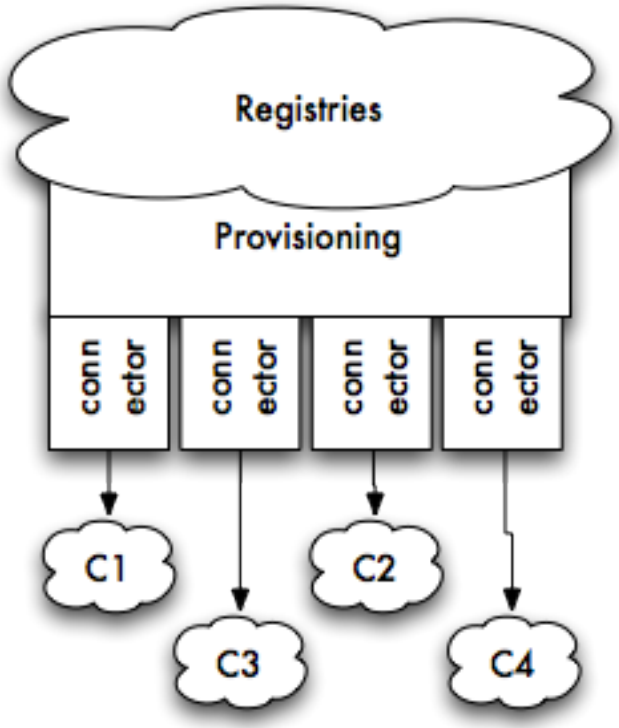
In the beginning...

- Simple component model from Chicago kick-off
- Registry (to hold the data)
- Access Management (to control the data)
- Provisioning (to distribute the data)
- Common Svcs (and in the darkness bind them



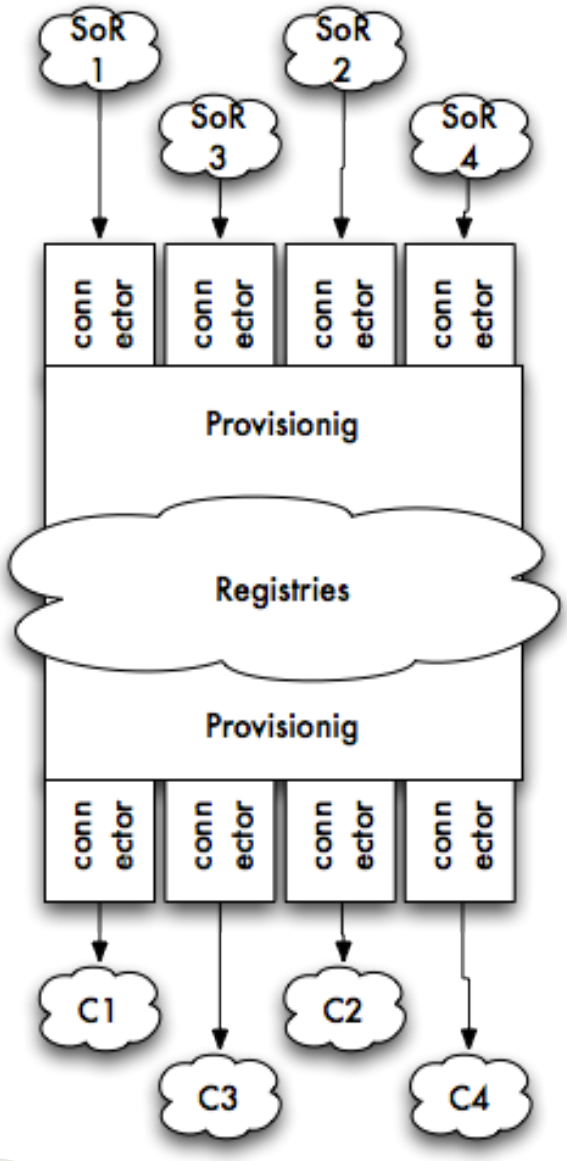
Provisioning: Initial Thoughts

- Provisioning = transport from Registry => Consumer
- Essentially, plumbing
- Transparent synchronization of attribute values R=>C
- Attributes become waste water; long, uninterrupted pipes



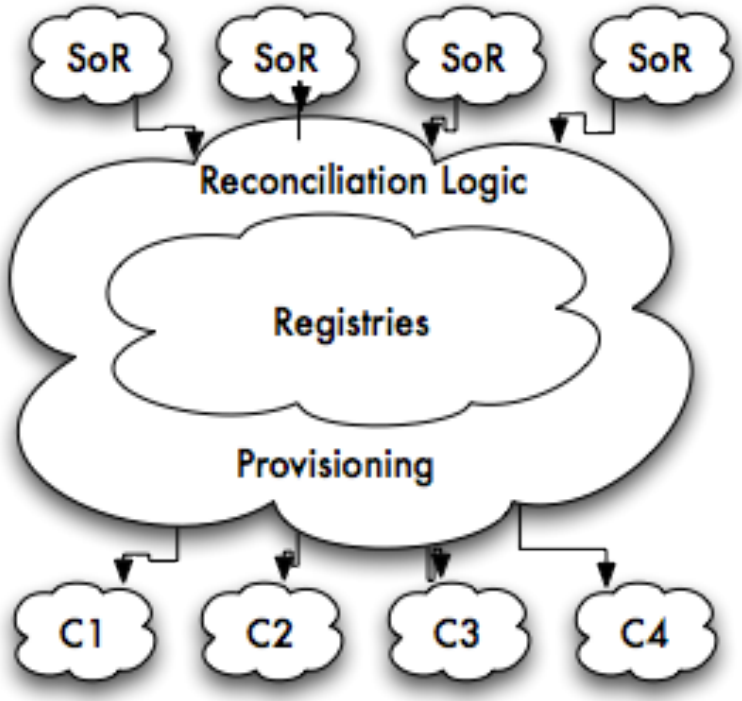
Evolution: Thoughts[2]

- Registry=>Consumer :: SoR => Registry
- SoR=>Reg < Reg => Cons (we need data to propagate)
- Plumbing, but both affluent and effluent (hopefully never confused)
- Still largely synchronization-centric, 1:1 attribute mapping



Evolution: Thoughts[3]

- Not transport -- **state** -- water quality, not plumbing
- Not synchronization -- **consistency**
- Provisioning == maintaining interentity state consistency
- Pipes are no longer passive; occasional clean-outs, chlorinators, pumps



Modeling State Consistency

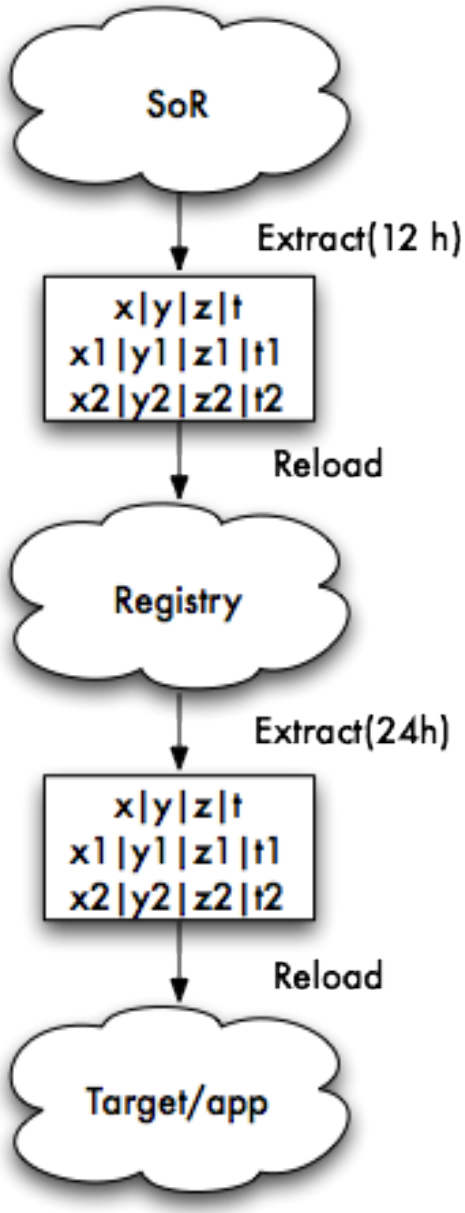
- Evolved view of provisioning starts to sound...familiar -- we realized there was prior art...
- Really just an instance of the broader class of Enterprise Data and Application Integration
- Well-defined patterns for EI (cf. EIP book & others)
- Consistency is a continuum -- similarity/T (long) to (transactional equivalence)/moments

Search for Strategies

- Obvious question: Which EI patterns apply, how, and how well?
- Spent time evaluating different strategies and talking through advantages and disadvantages, hoping for “clear wins”…

Periodic Batch Refresh

- A source (SoR or Registry) periodically produces full extract containing all relevant attribute values for all relevant entries. A consumer (Registry or target app) reloads its identity store/cache entirely from the extract.

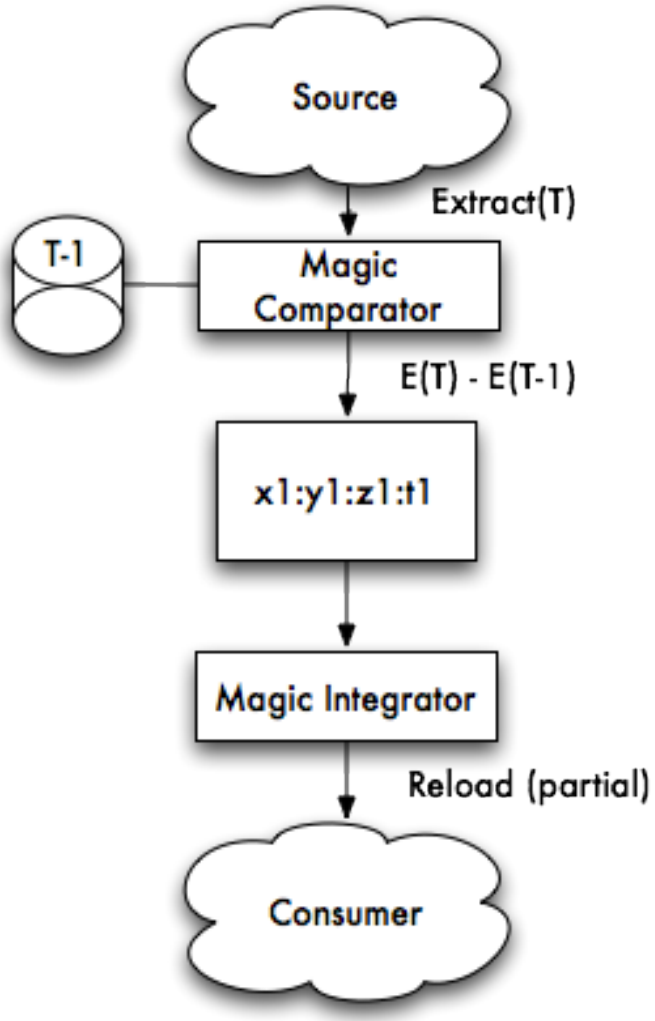


Periodic Batch Refresh

- Advantages
  - Easy
  - Predictable in time
  - Self-healing
  - Automatic impedance matching
- Disadvantages
  - Latency, Execution cost
  - What's an Audit?
  - Secondary transactional effects may be lost
  - Source state sensitivity
  - Unidirectional; JIC only

Periodic Differential Refresh

- A full extract of relevant source data is delivered to a magical comparator that computes differences between the extracted state (T) and a previously stored state (T-1). Differences computed are applied as a partial reload (still overwriting entries) in the consumer.

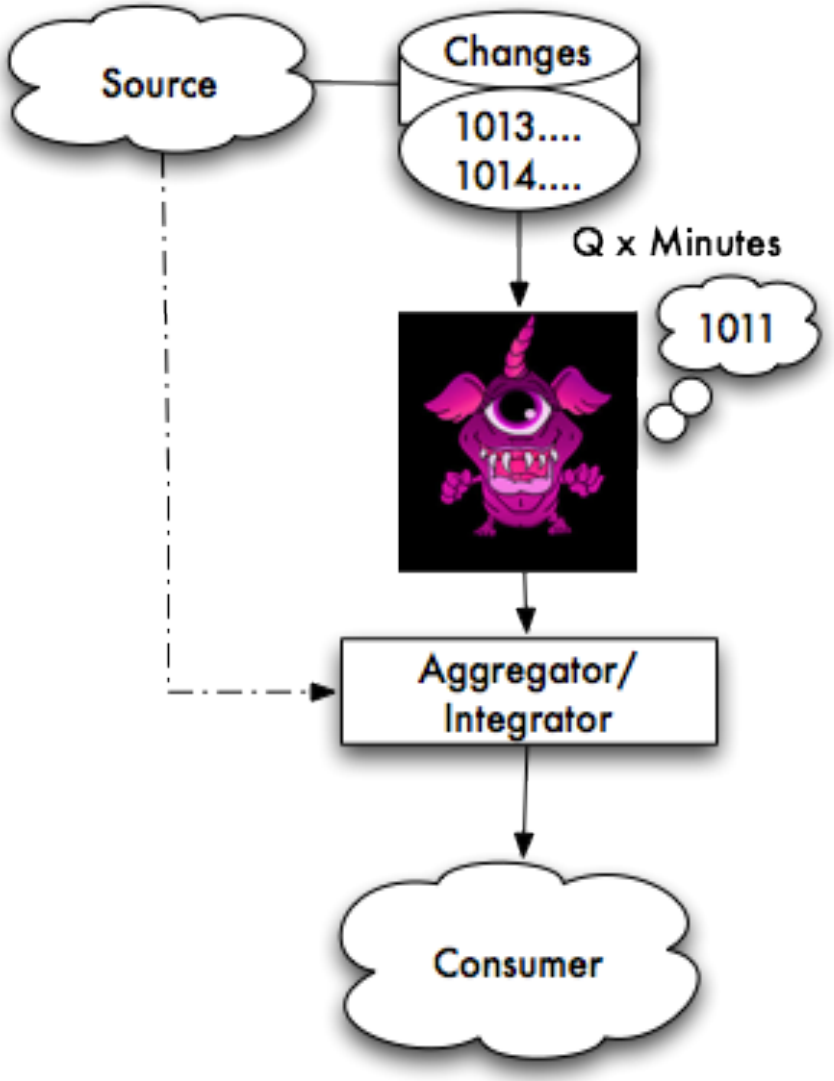


Periodic Differential Refresh

- Advantages
  - Relatively simple
  - Predictable in time
  - Cheaper for consumer; bandwidth sparing
  - Gross historical audit
  - Potentially self-healing (persist old state on fail)
- Disadvantages
  - High latency
  - Runtime expense
  - Sensitivity to source state
  - Brittle/policy changes
  - May miss secondary Tx consequences; uni-directional; JIC

Changelog Consumption[2]

* Source generates a changelog , assigning monotonically increasing IDs to changes and recording them sequentially. A purple changlog eater tracks its last successful change ID and periodically “rolls forward” changes to aggregator. Changelog entries may be “heavy” (with all associated data) or “light” (aggregator has to retrieve changed data)

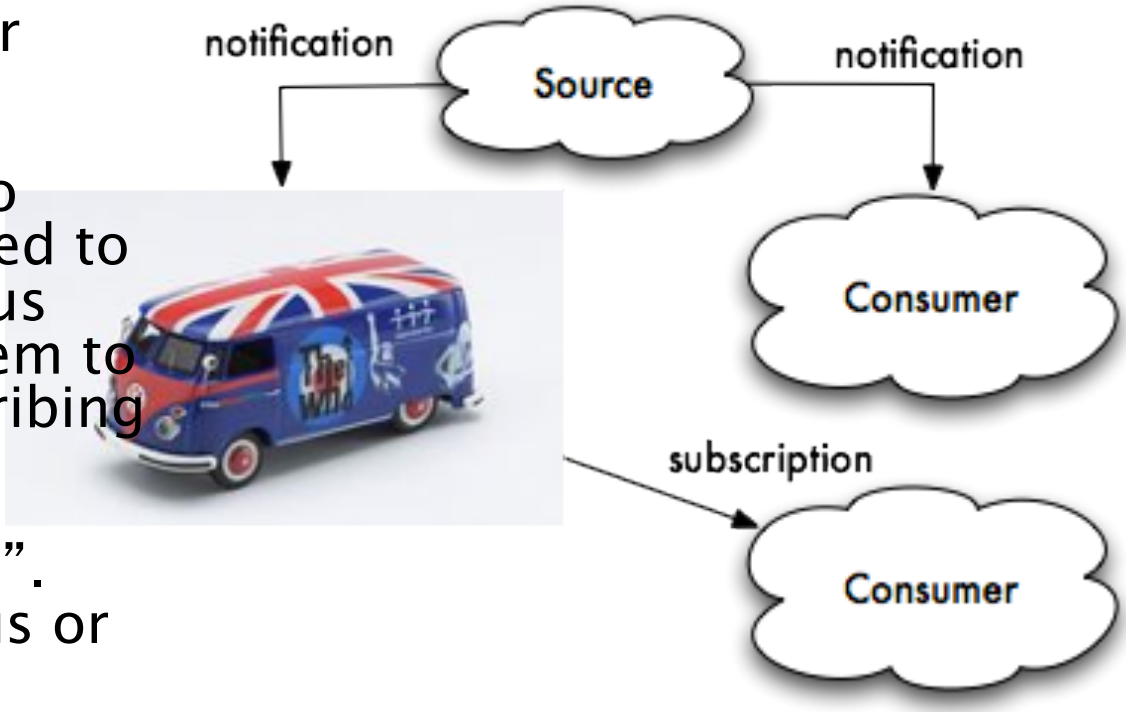


Changelog Consumption[2]

- Advantages
  - Reduced latency
  - Relatively cheap runtime
  - Potentially self-healing (if the PCE memory is good)
  - Very auditable
  - Tx integrity assurable
- Disadvantages
  - Still, guaranteed latency
  - Somewhat brittle/policy
  - We don't make no steenkin' changelog
  - Impedance mismatch - atomic vs. composite ops source vs. consumer

Messaging[n]

* Source generates synthetic notification messages whenever changes occur. Messages may be delivered directly to consumers or passed to a (magic) service bus which may pass them to one or more subscribing consumers. Often implemented as “changelog + push”. May be synchronous or asynchronous.



Messaging[n]

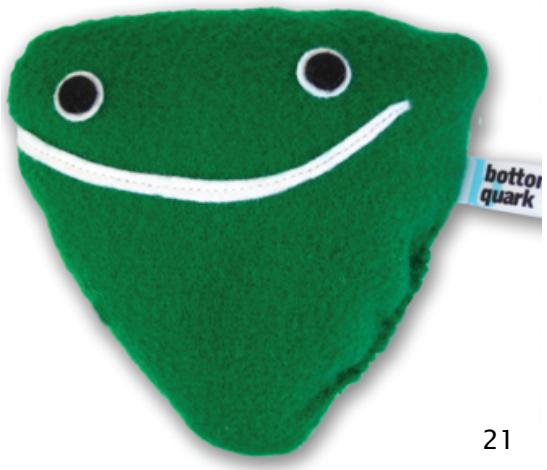
- Advantages
  - lim(latency) --> 0
  - cheap at the ends
  - Easily audited
  - Resilient / policy change (with ESB)
  - Strong for JIT provisioning
- Disadvantages
  - expensive in the middle
  - Messages? Standards?
  - Brittle/failures (by itself)
  - Tx's must still map ; mismatched atomicity can cause instability

Integrative Chromodynamics

- Top(down) approach
  - Find real use cases
  - Design solutions of each strategy, recording req's
  - Analyze minimal interface "knobs"
  - Identify requirement spanning set
  - Define API covering it
  - Analyze gaps in extant solutions; bridge them as necessary with code.

Integrative Chromodynamics

- Bottom (up) approach (from Delivered APIs)
  - Tabulate APIs from OR, CPR,…
  - Set up functional categories
    - Reflect, Reconcile, Credential, Enrich, Provide
  - Sort APIs into functional bins
  - Look for overlaps, gaps, singletons
  - Integrate with top (down) findings & get to work

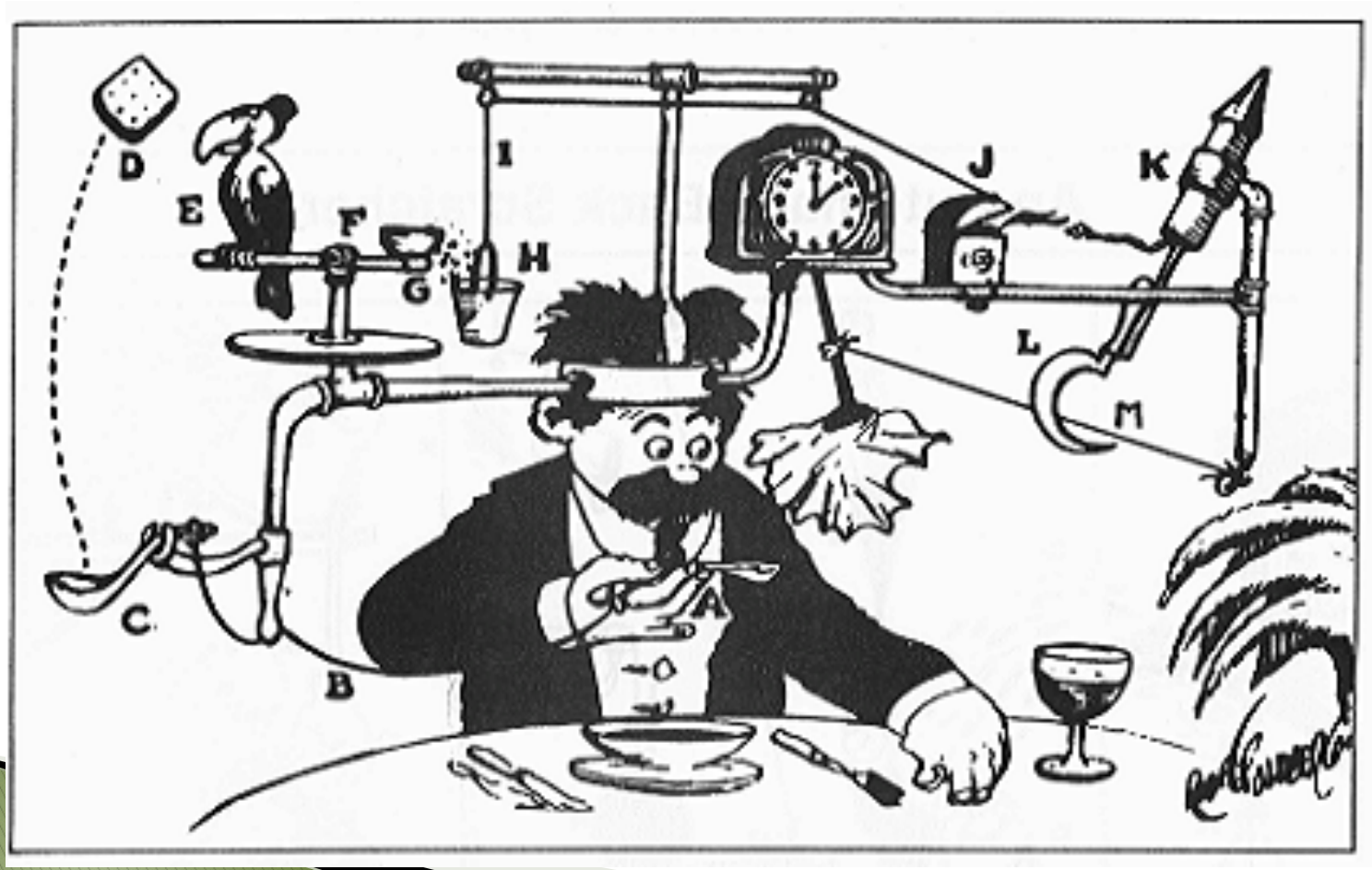


P & I The Anti-product

- It's the one IT thing you can't pay people to take away. It's worse than the garbage that way

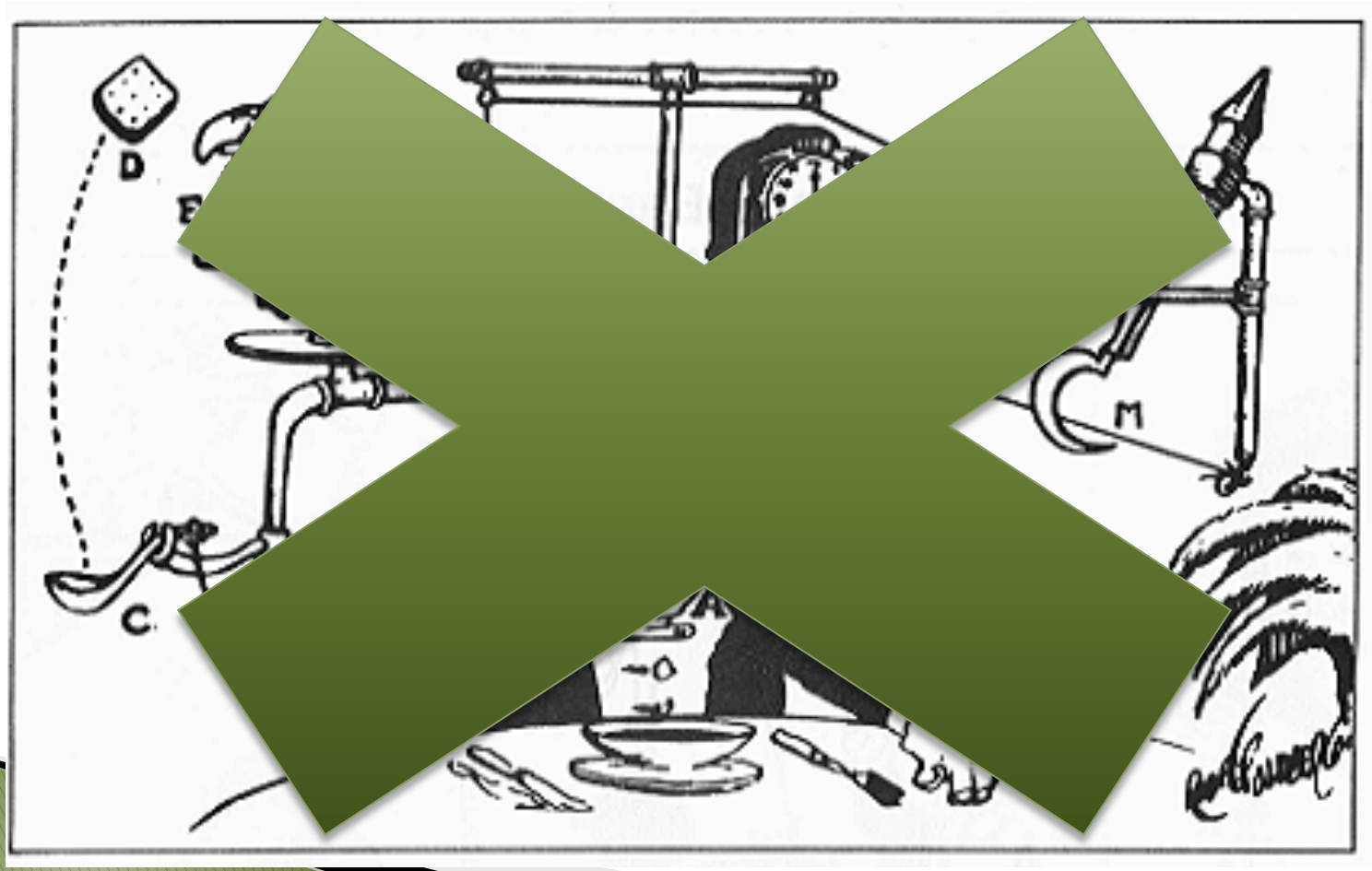
P & I The Anti-product

- "here is your P&I solution -- install, execute, PROFIT!"



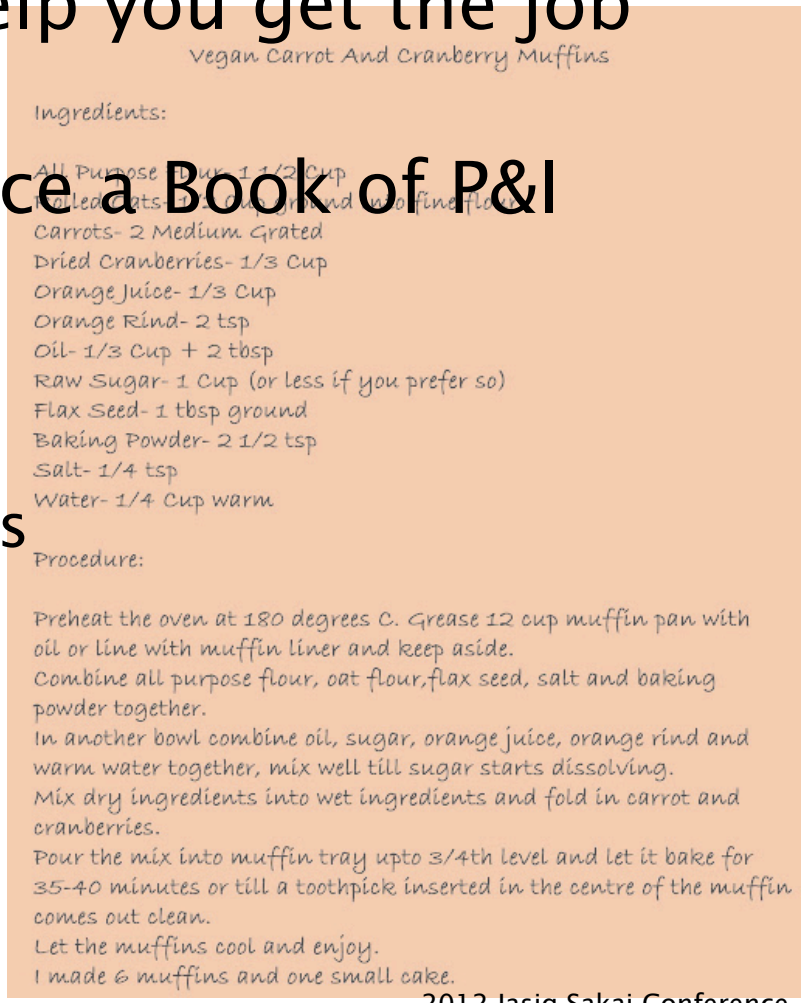
P & I The Anti-product

- "here is your P&I solution -- install, execute, PROFIT!"



Late-night Epiphanies re P&I

- 1) There are many strategies or models (see above), each one can help you get the job done
- SO, CIFER should produce a Book of P&I Recipes:
  - Common dishes
  - Common ingredients
  - With optional substitutions



Vegan Carrot And Cranberry Muffins

Ingredients:

All Purpose Flour- 1 1/2 Cup
Rolled Oats- 1/2 Cup ground into fine flour
Carrots- 2 Medium Grated
Dried Cranberries- 1/3 Cup
Orange Juice- 1/3 Cup
Orange Rind- 2 tsp
Oil- 1/3 Cup + 2 tbsp
Raw Sugar- 1 Cup (or less if you prefer so)
Flax Seed- 1 tbsp ground
Baking Powder- 2 1/2 tsp
Salt- 1/4 tsp
Water- 1/4 Cup warm

Procedure:

Preheat the oven at 180 degrees C. Grease 12 cup muffin pan with oil or line with muffin liner and keep aside.
Combine all purpose flour, oat flour,flax seed, salt and baking powder together.
In another bowl combine oil, sugar, orange juice, orange rind and warm water together, mix well till sugar starts dissolving.
Mix dry ingredients into wet ingredients and fold in carrot and cranberries.
Pour the mix into muffin tray upto 3/4th level and let it bake for 35-40 minutes or till a toothpick inserted in the centre of the muffin comes out clean.
Let the muffins cool and enjoy.
I made 6 muffins and one small cake.

Late-night Epiphanies re P&I

- 2) As always, the right tool/utensil makes the job easy
- SO, Identify good, useful tools (or create them IFF necessary)
- Describe their uses

Late-night Epiphanies re P&I

- 3) Show is always better than tell
- SO, Actually bake some cake, saute some morels or cook a goose (or not)
- More “Sample Solutions”
- Less “Reference Implementations”

CIFER Provisioning and Integration

Don’t fall for the RonCo IntegratoMatic pitch

!!

!!

!!

!!

CIFER Provisioning and Integration

- A Recipe Book
- Toolkits
- Sample Solutions

- We can't do it for you (no one can)
- But we ***CAN***, maybe, help you succeed at DIY

Yet More CIFER goodness at Jasig

- *Open Source Person Registries – You want 'em, we got 'em!*
  - Dedra Chamberlin, UC Berk, Eric Westfall, UI
  - 13-Jun-2012
  - 2:30 PM – 3:30 PM
  - Here (Conference Center Room 3 (7th Floor))

We certainly raised more questions than we answered. Ask away:

For More Information:

info@ciferproject.org

http://ciferproject.org

CIFER

Community Identity Framework for Education and Research