

Gartner Business Transformation & Process Management Summit
London March 2016

Riva

An exercise in process design: a restaurant

Martyn Ould



Copyright © Martyn Ould 2016

Openers



- /// Why are you here?
- /// What is it about processes that you are most interested in?
- /// What methods are you using now?
- /// What problems do you find/anticipate?

Anyone care to define a
'process'?



The first challenge

- /// Deciding what processes we have
- /// ... in a way that
 - /// yields 'true' processes based on our business
 - /// recognises the concurrency of activity
 - /// is not affected by 'design decisions' such as organisational structure, technology, culture

A clever move:

- /// In Riva we don't bother trying to define the word/concept 'process'
- /// We simply have a way of determining what processes an *organisation* in a particular *business* must have
- /// 'If you are in *this business* you must have *these processes*'

Anyone care to suggest
how to decide what
processes we have?



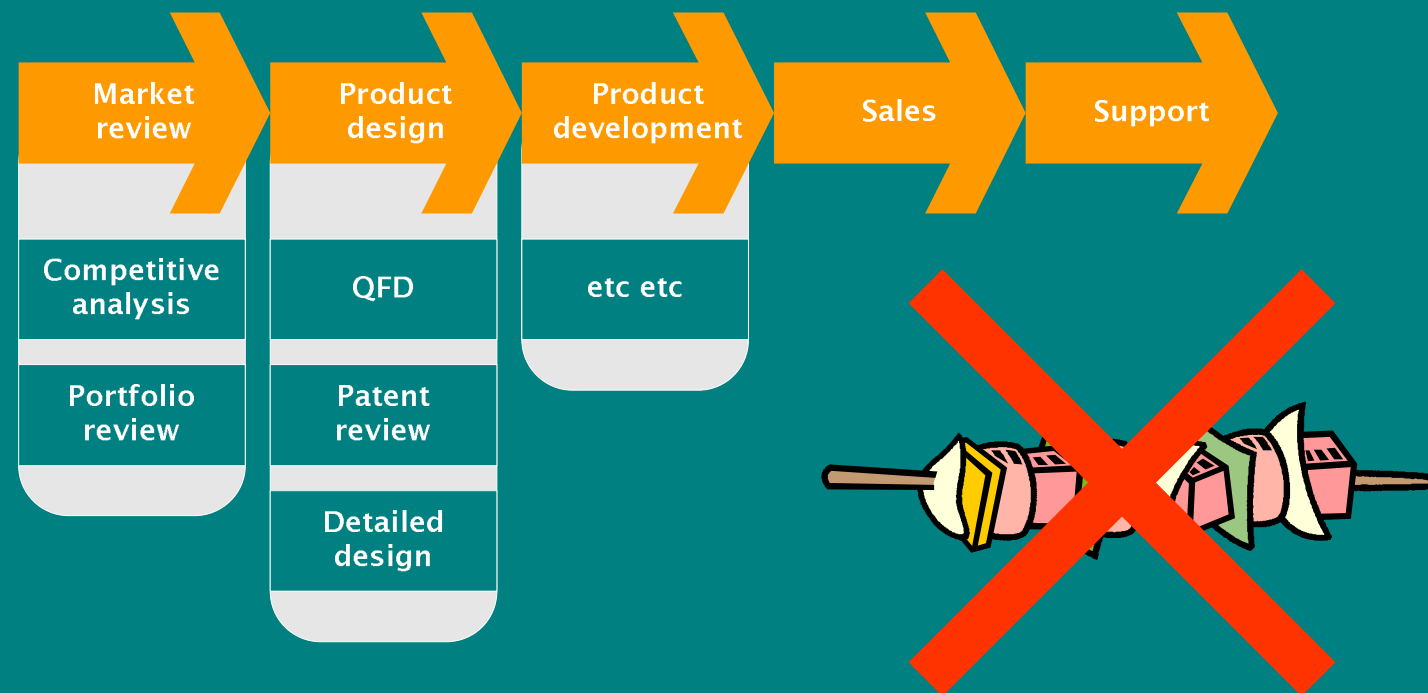
How not to decide on your processes – the axe

- /// Your world has structure: you can't just ignore it and chop where you like



How not to decide on your processes – the kebab

/// The world is not simple and serial



How not to decide on your processes – the organogram

- /// The organogram is just today's choice of structure – a design decision



Dangers when thinking about the dynamics of an organisation

- /// Assuming things are sequential
- /// Ignoring the possibility of concurrent activity
- /// Thinking of things purely from the point of view of the current organisational structure
- /// Thinking of things purely from the point of view of the current technology or culture
- /// Thinking purely in terms of the flow of stuff
- /// Thinking purely in terms of inputs and outputs

Organisations are



- /// highly concurrent
- /// highly collaborative

- /// Any method for thinking about processes must capture this

The *Riva* method contains techniques for

- /// preparing the organisation's *Process Architecture Diagram*
- /// mapping a process on a *Role Activity Diagram* for
 - /// understanding or definition
 - /// diagnosing problems
 - /// examining the relationship between organisation and process
 - /// designing a process
 - /// defining requirements for BPMS/IS support
- /// in a way that captures all the concurrency and collaboration

A *Riva* process architecture shows

- /// all the processes involved in the conduct of the business
- /// and their dynamic relationships

- /// Let's build the process architecture for our restaurant

Let's start by characterising our business

'If you are in *this business* you must have *these processes*'



We do this by listing its *essential business entities*:

those *things* that are the *essence* of the *business*
of the organisation concerned
– in our case a restaurant

Some EBEs are special

- /// Some require *work* to look after them during their lifetimes
- /// We call them *Units of Work* – for example:
 - a customer order
 - a customer
 - a clinical trial
 - a planning application
 - a patient visit
 - a product
 - a plant failure

Let's check which of our EBEs are *units of work*



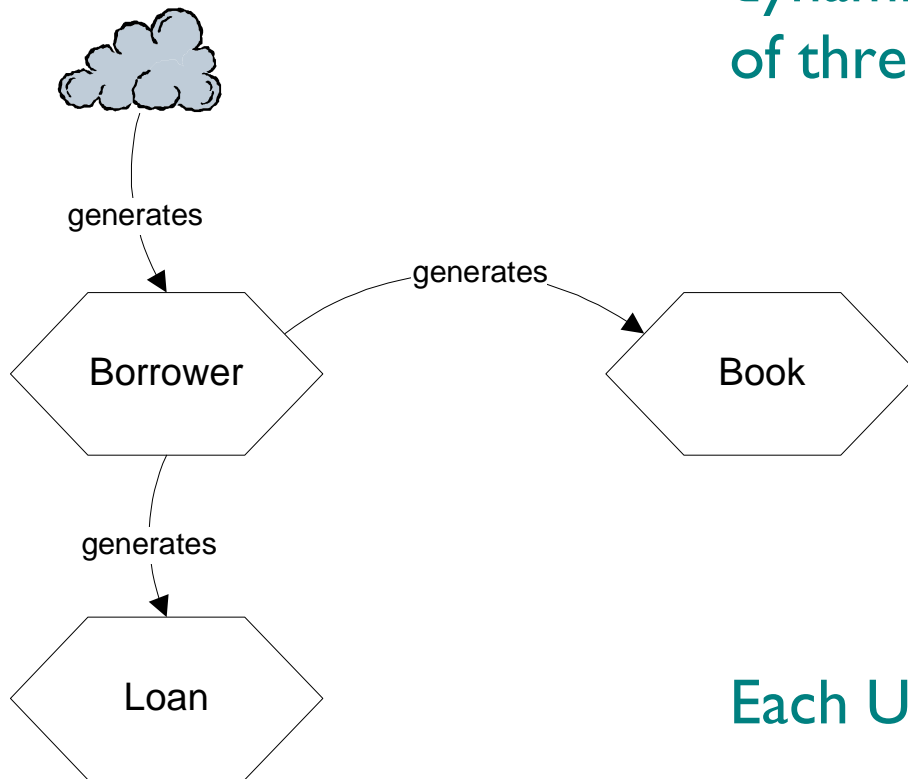
- /// In other words, which of them have lifetimes during which we must look after them?

As the organisation runs, units of work multiply

- /// A drug compound *generates* (requires) clinical trials
- /// A clinical trial *generates* (recruits) patients
- /// A plant failure can *generate* (require) maintenance tasks
- /// A planning application can *generate* (require) site inspections
- /// An occupied table *generates* orders for dishes

We capture the business dynamics in a Unit of Work Diagram

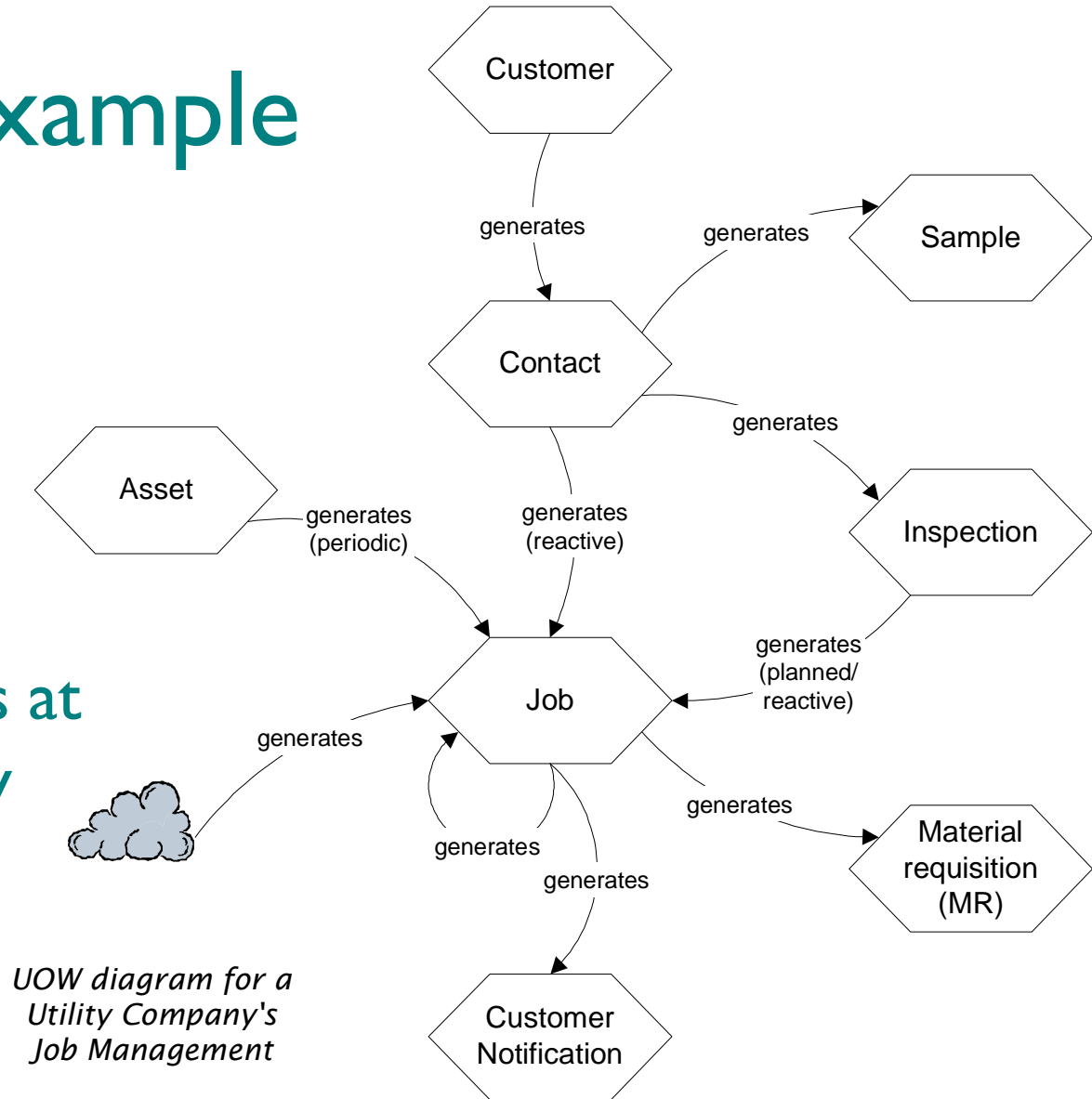
This diagram captures the dynamics of a library in terms of three units of work



Each UoW has its own lifetime

Another example

The business of carrying out jobs at a utility company

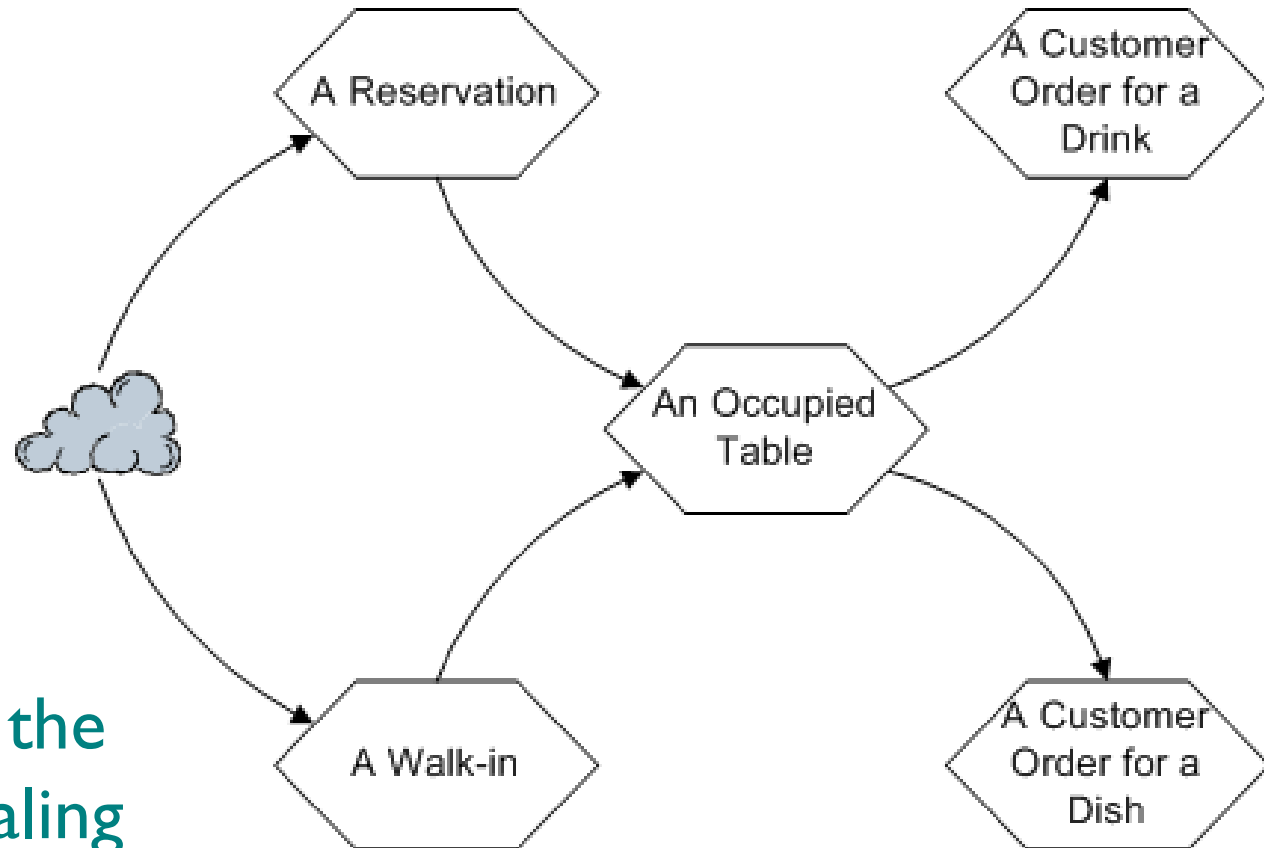


Identify the dynamic relationships between our units of work



- /// Which units of work during their lifetime can *generate* which other units of work?

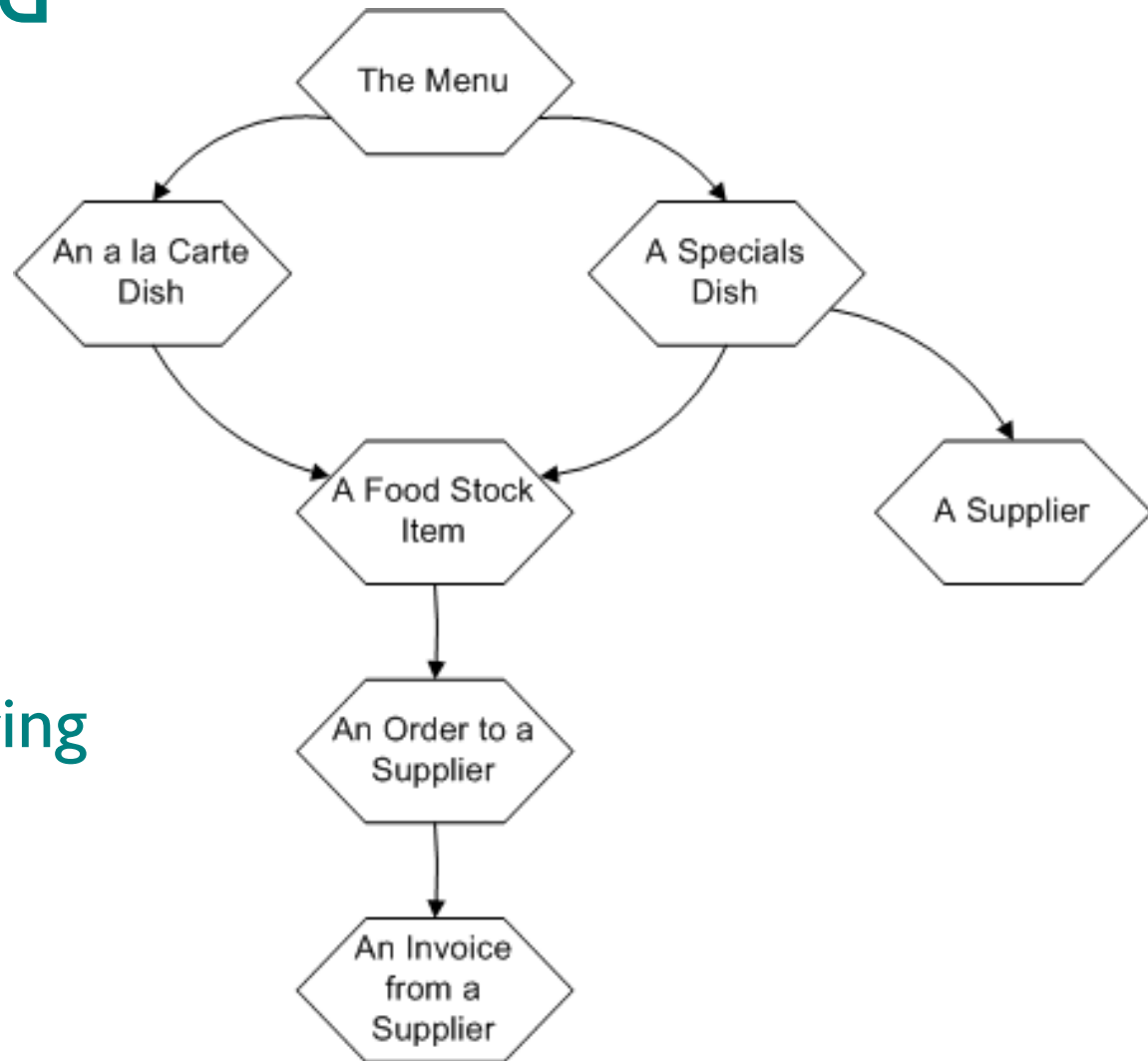
Unit of Work Diagram for the 'foreground'



The part of the
business dealing
with customers

Unit of Work Diagram for the 'background'

The part of the
business dealing
with the
restaurant's offering
to customers



How do we get from units of work to processes?

Units of work require processes



- /// We need a process to handle each unit of work:
 - 'Handle a customer order'
 - 'Handle a customer'
 - 'Handle a clinical trial'
 - 'Handle a planning application'
 - 'Handle a plant failure'
- /// We call this the *case process*
- /// It is instantiated whenever a new case is generated
- /// It deals with the *lifetime of the case*

Let's list the case processes of our restaurant



The cases of a unit of work require management



- /// We must schedule, prioritise, and resource the competing cases
- /// That's the job of the *case management process*
- /// Examples:
 - /// 'Manage the flow of clinical trials'
 - /// 'Manage the flow of maintenance tasks'
 - /// 'Manage the flow of site inspections'
 - /// 'Manage the flow of food orders'

Important note:

- /// At any one time there can be many instances of the *case process* in progress
- /// But there is only one instance of the corresponding *case management process*, looking after them all and receiving requests for new case process instances

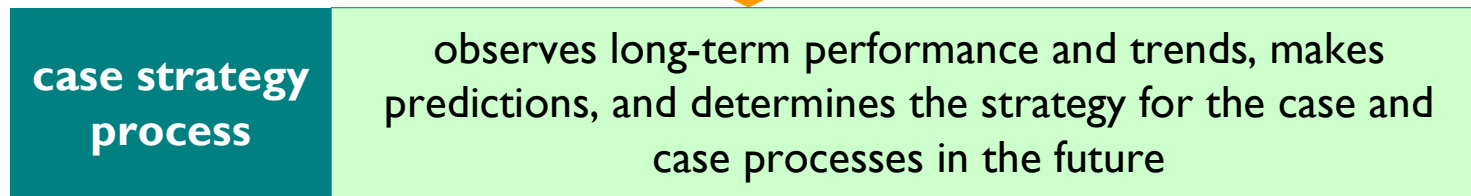
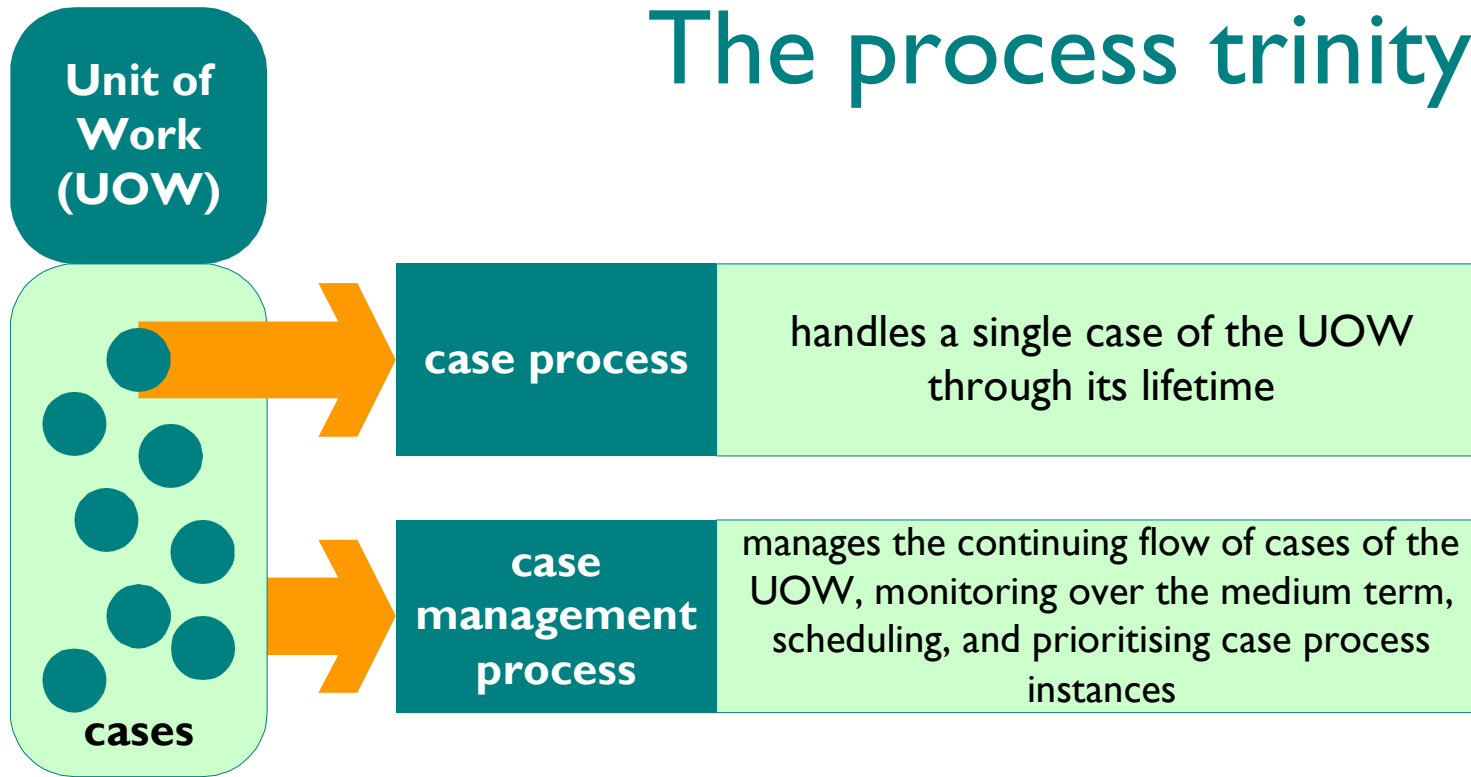
Let's list the case management processes of our restaurant



The third process type

- /// The *case strategy process* responds to trends and environmental changes
- /// For example:
 - /// 'Take a strategic view of customer calls'
 - /// 'Take a strategic view of plant failures'
 - /// 'Take a strategic view of planning applications'
 - /// 'Take a strategic view of a la carte dishes'

The process trinity



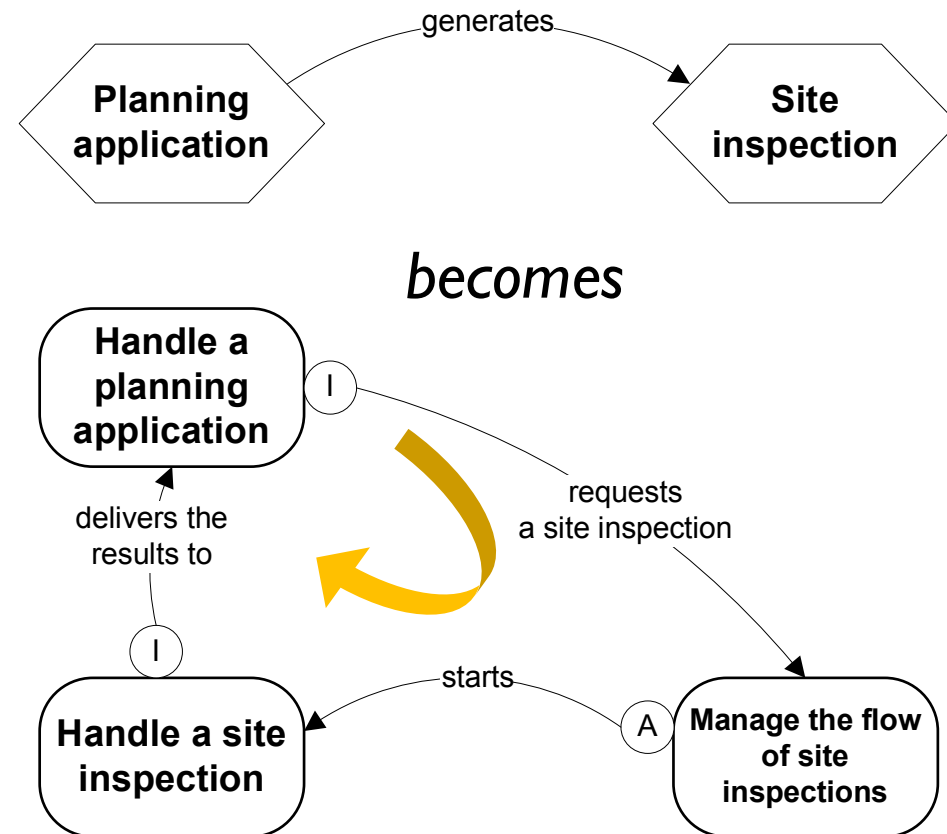
How do we get from the
processes to their relationships?

The process architecture can initially be derived directly from the UoW Diagram

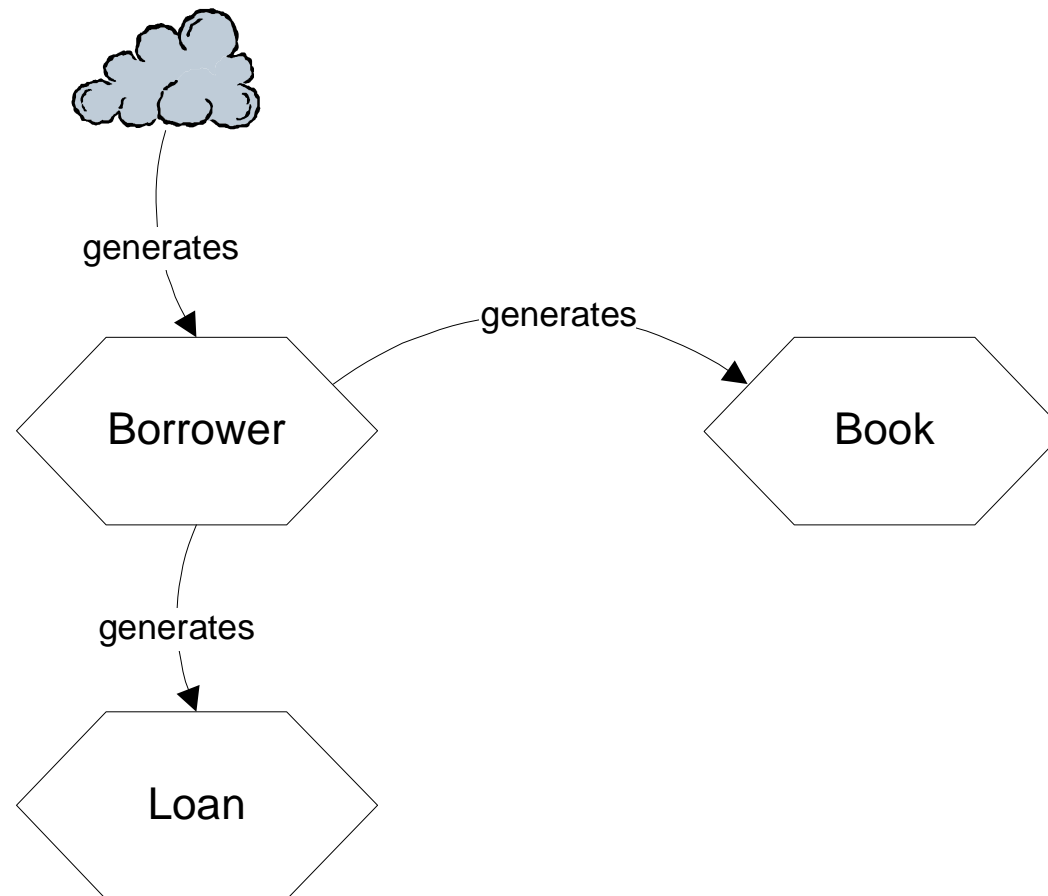


Each 'generates' relationship between two units of work implies a set of relationships between their case and case management processes

We call this the 'first-cut' architecture

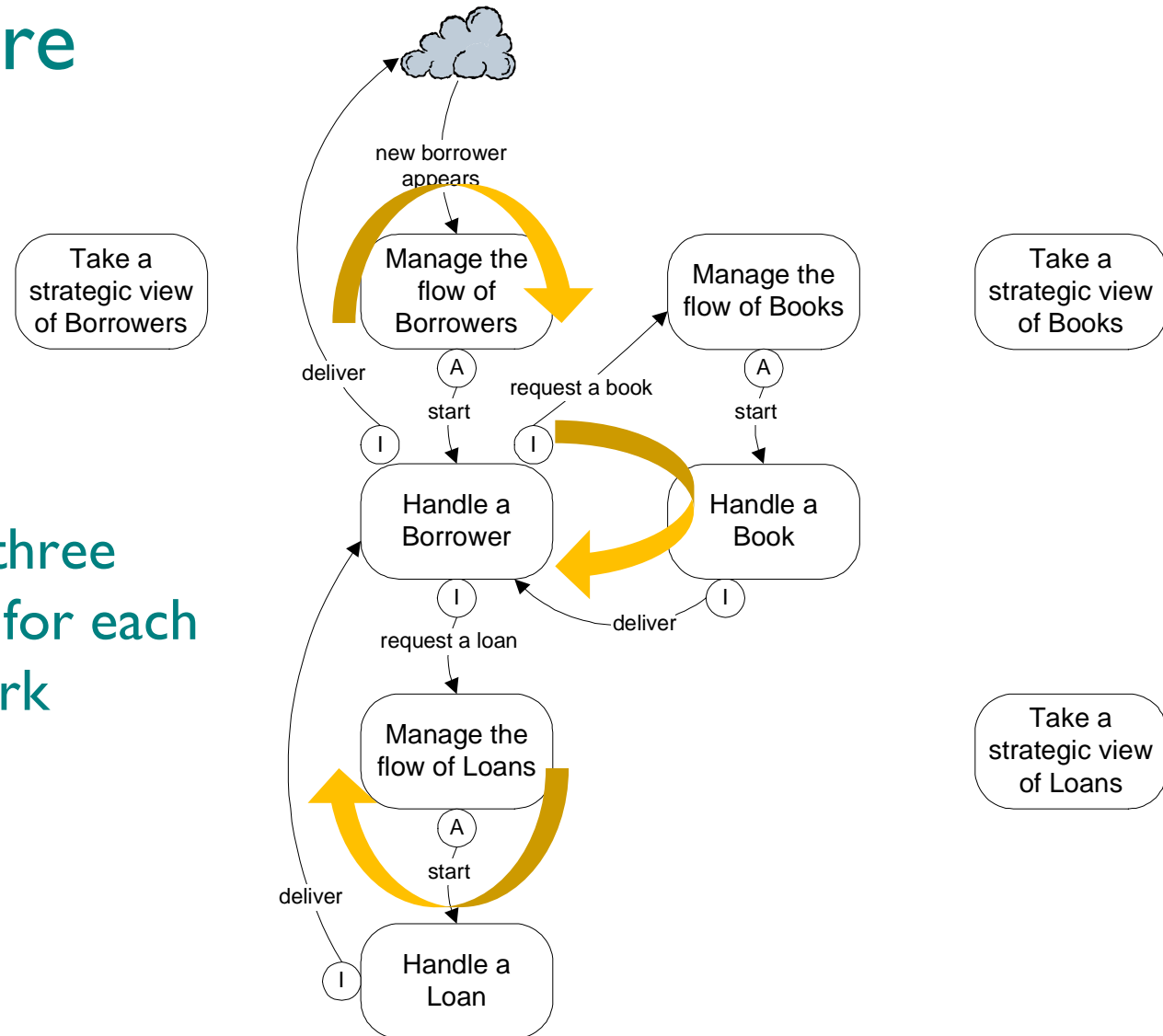


The three UoWs of the library . . .

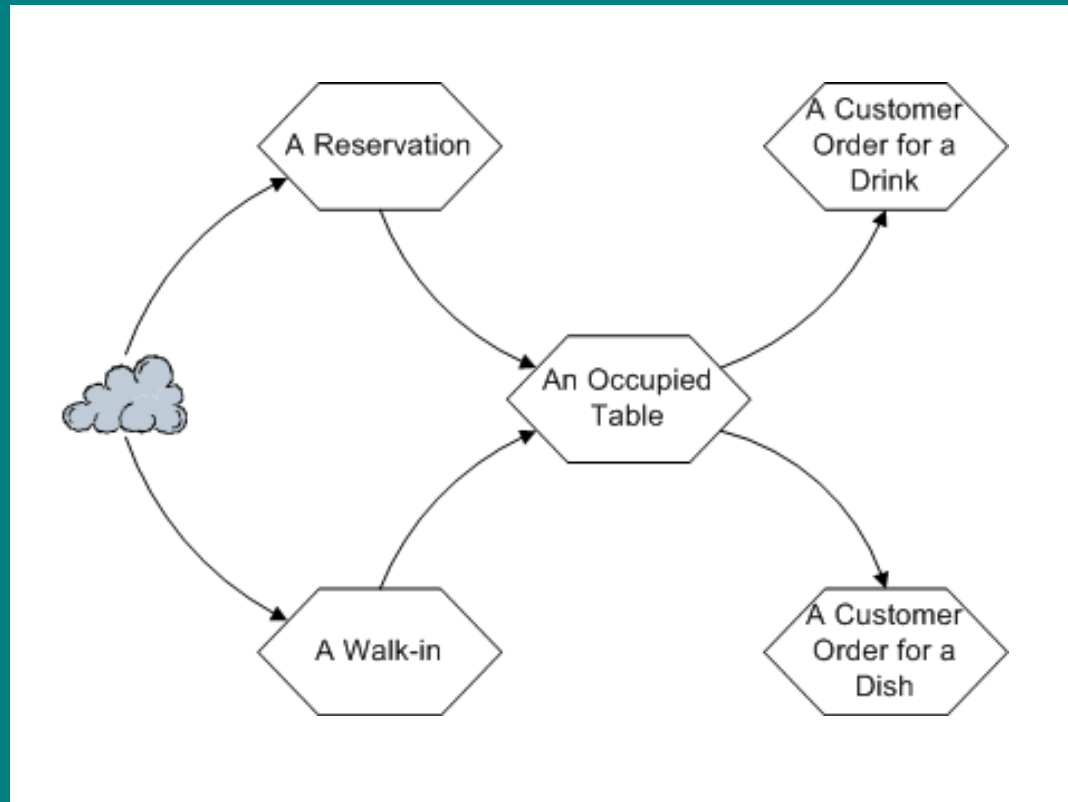


... generate this 'first-cut' process architecture

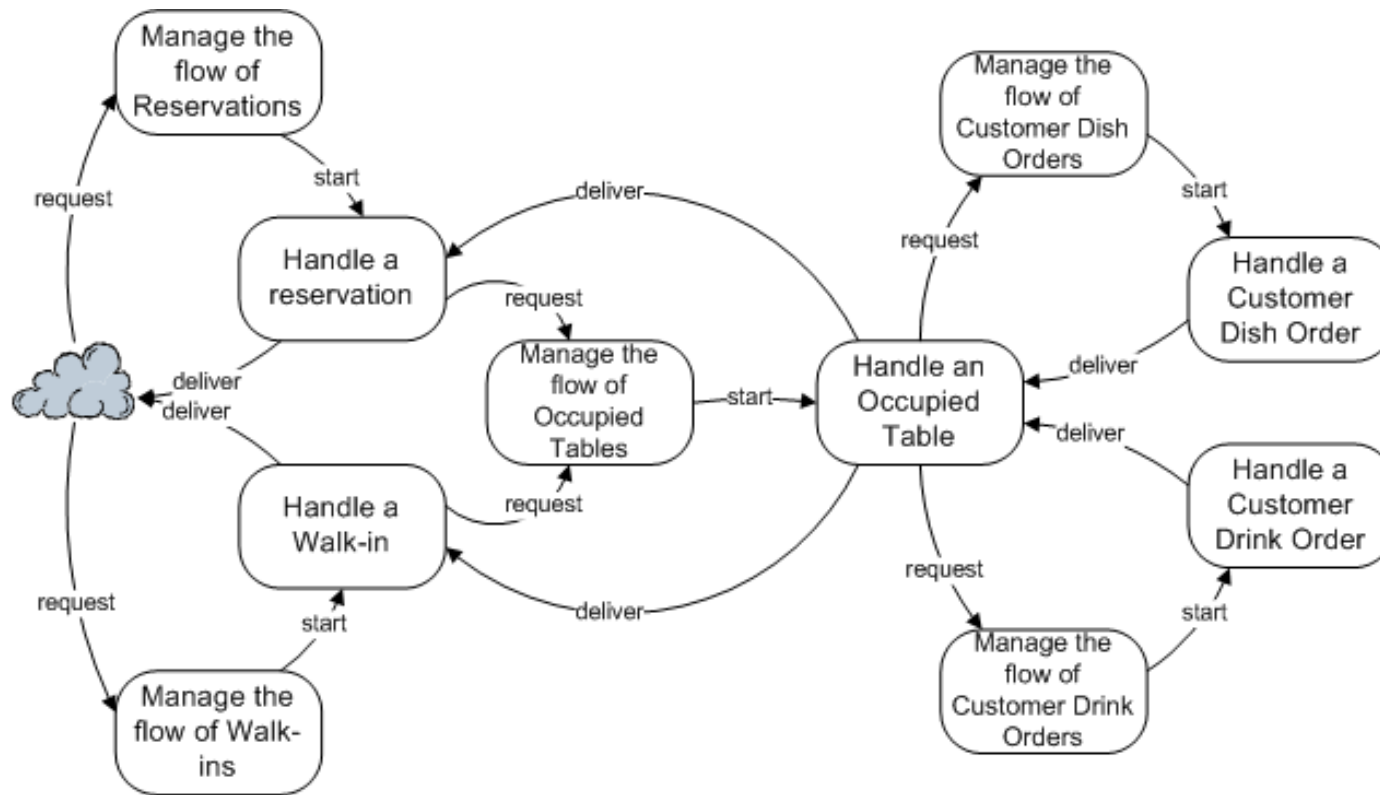
/// Note the three processes for each unit of work



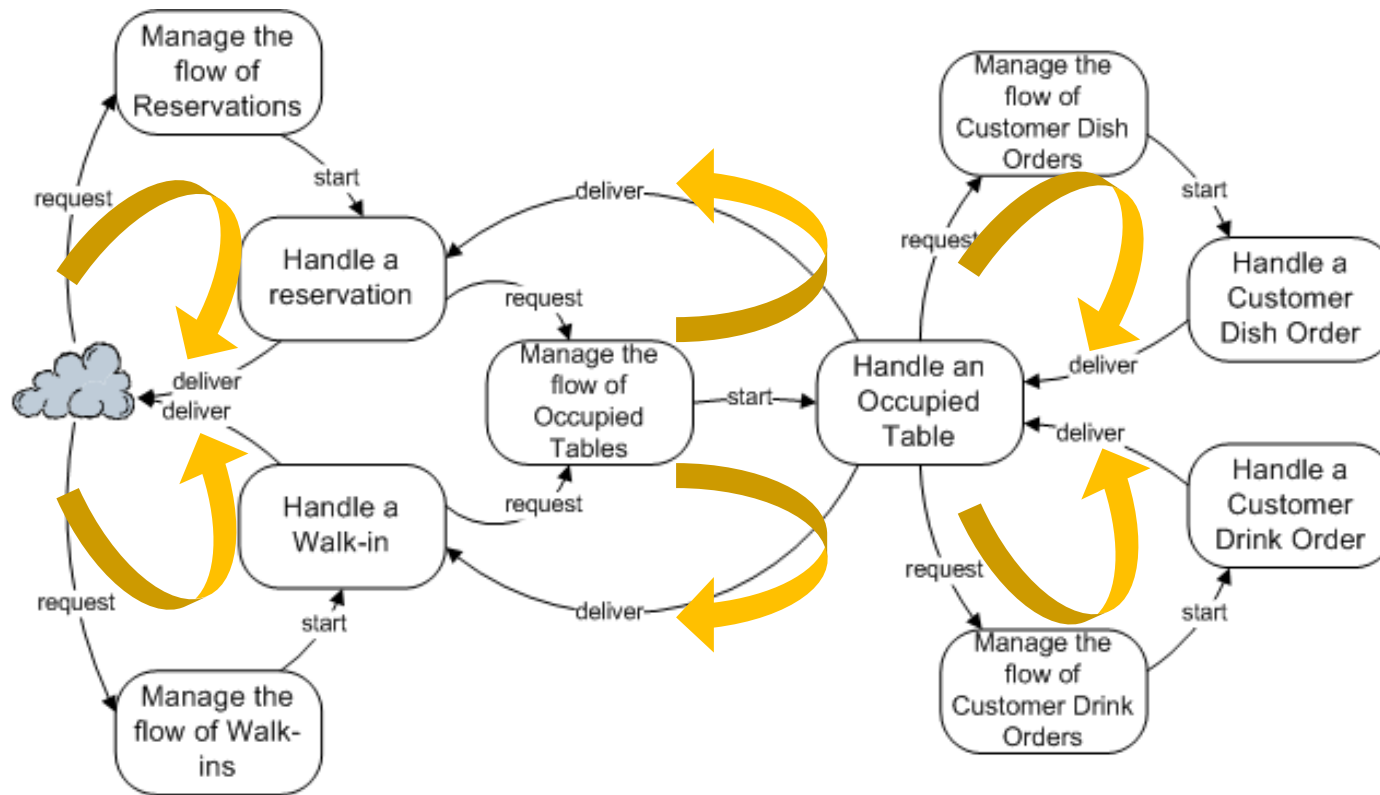
Sketch the first-cut architecture for this fragment



The first-cut architecture for this fragment



The first-cut architecture for this fragment



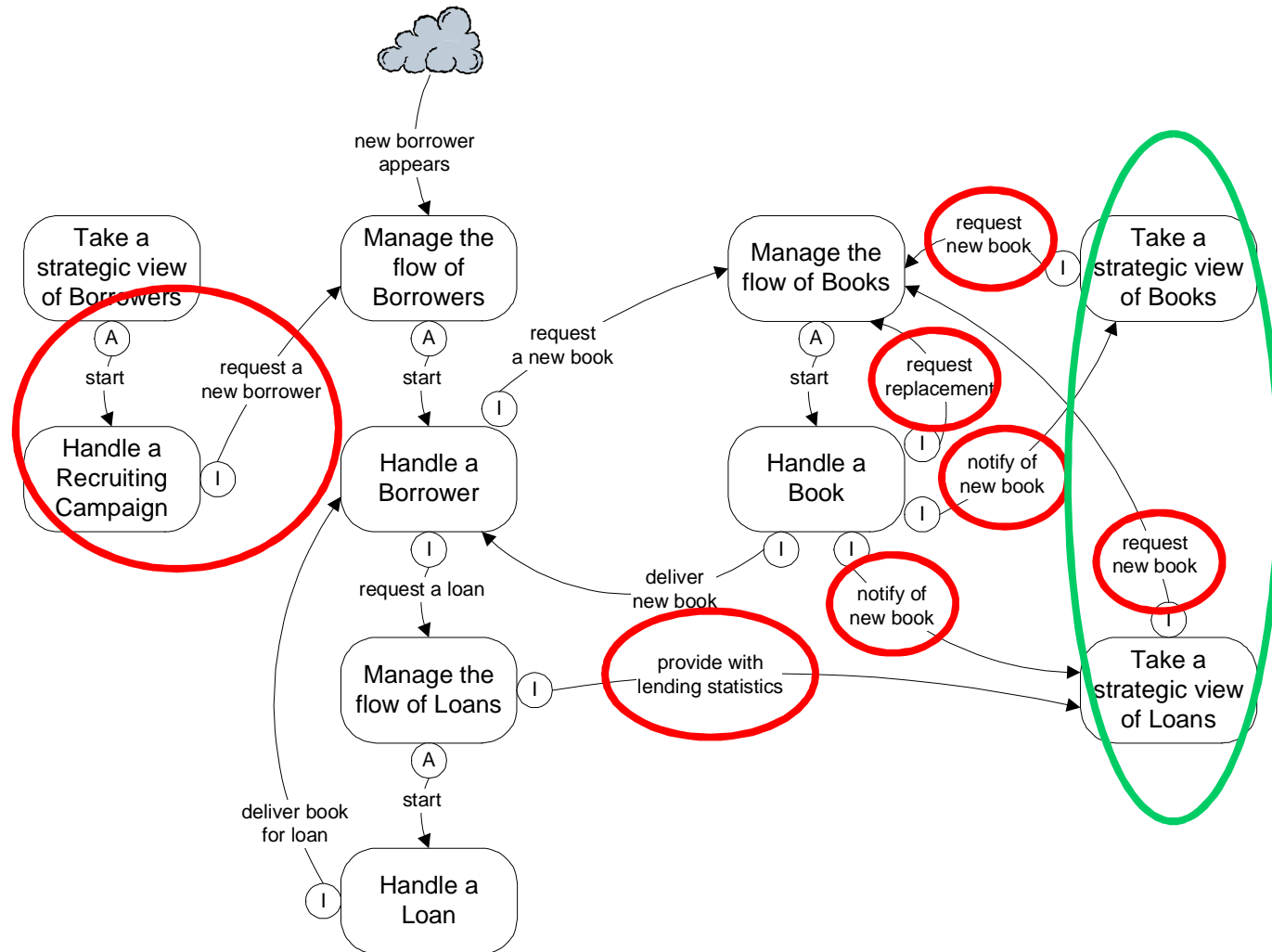
That was mechanical – is the result true to actuality or to our desires?

- /// We now put the first-cut process architecture 'against the world'
- /// We can apply certain refinements to reflect reality

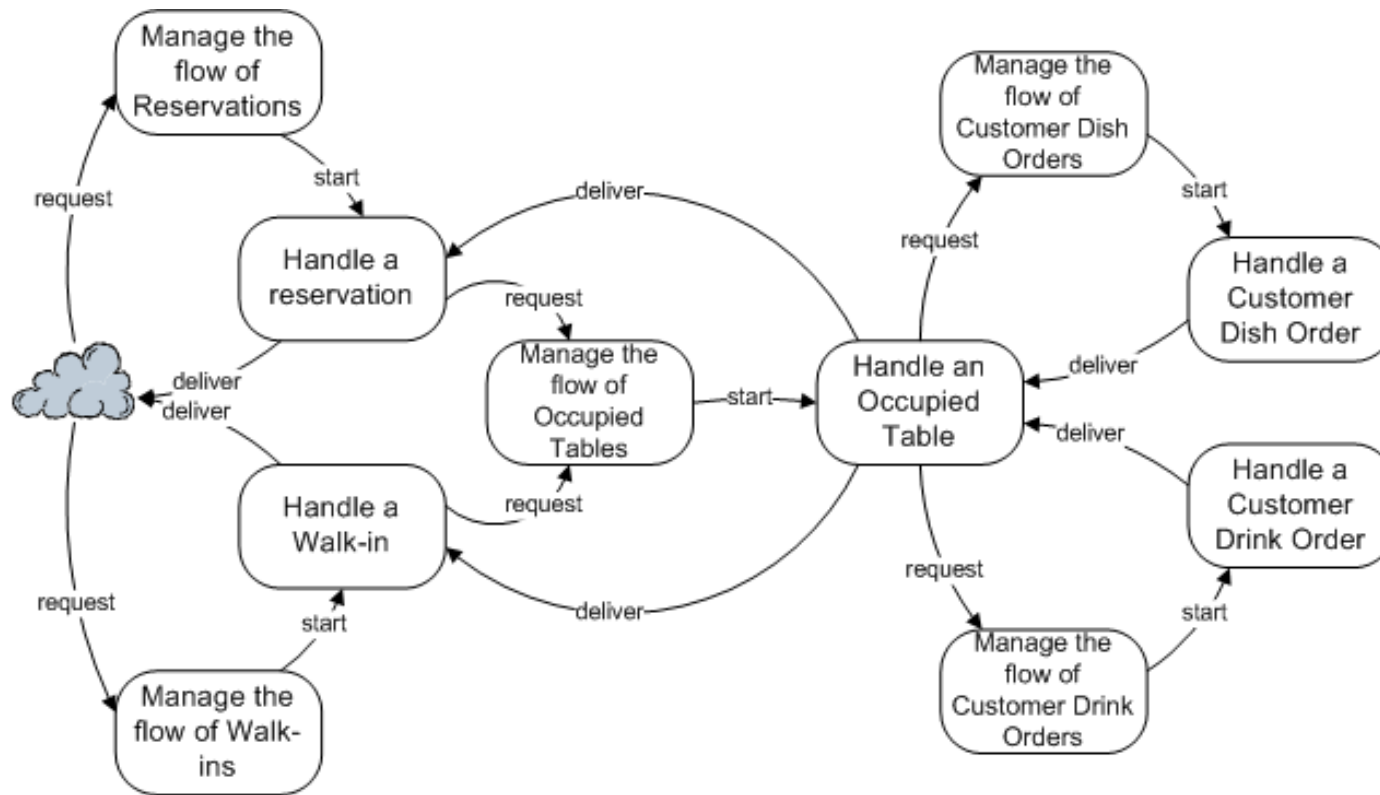
The second-cut process architecture

- /// We can introduce *design decisions* about how we want things to work
- /// We can remove processes that are redundant or empty in our situation, especially case management processes
- /// We can remove process interactions that are redundant or empty in our situation
- /// We can combine processes, especially case strategy processes
- /// We can introduce ‘designed units of work’ and their processes
 - /// invoice, monthly report, complaint, batch, survey

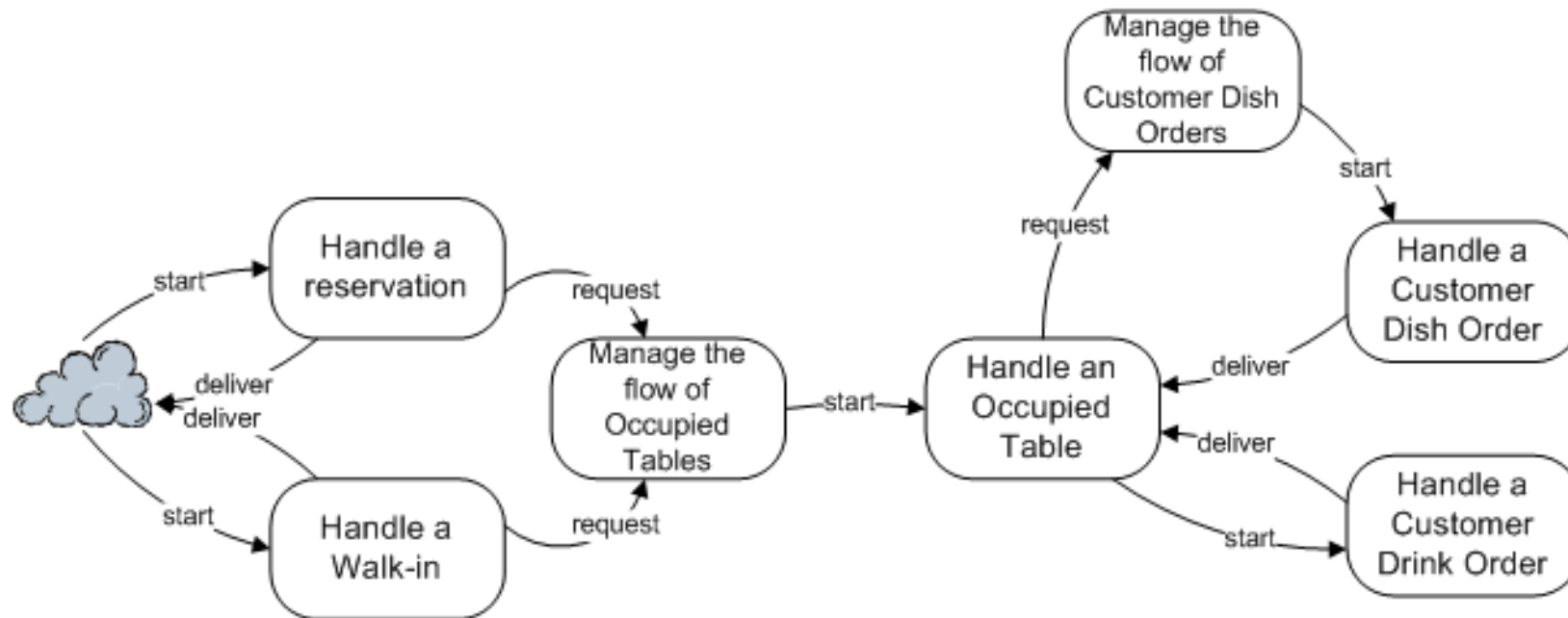
Refinements for the library's second-cut process architecture



What refinements could we make here?



My restaurant



Riva's five steps to the process architecture for any organisation



1. Brainstorm the *essential business entities*
2. Extract the *units of work*
3. Prepare the *Unit of Work Diagram*
4. Transform into the 'first-cut' *Process Architecture Diagram*
5. Refine this into the 'second-cut' *Process Architecture Diagram*

The second challenge

- /// Producing a process model that is useful for our purpose
 - /// for understanding
 - /// for instruction/training
 - /// for process design
 - /// for process improvement
 - /// for organisation design
 - /// for IS specification

Dangers when thinking about an individual process

- /// Assuming things are sequential
- /// Ignoring the potential of concurrent activity
- /// Thinking of things assuming the current organisational structure, technology or culture
- /// Thinking just in terms of the flow of stuff
- /// Thinking just in terms of inputs and outputs

Riva process models use two central concepts

/// Role

/// where responsibilities are carried out

/// Interaction

/// where collaboration happens

Identify some roles in our restaurant



Roles

- /// A *Riva* process model chunks a process into a network of collaborating roles
- /// Each role is a business responsibility
- /// A role can
 - /// be permanent and ‘concrete’: e.g. Head of R&D
 - /// or be transient and ‘abstract’: e.g. the responsibility for safety on a project
 - /// have several active instances: e.g. the responsibility for a project
- /// Role instances interact to collaborate

Three different concepts – an example



- /// A role type: *MP (a generic post)*
- /// A role instance: *MP for Witney*
- /// An actor: *David Cameron*, currently acting the role instance *MP for Witney*
- /// He is also acting the role instance *Prime Minister of the UK*, one of many instances of *Prime Minister*
- /// Elections are the mechanism by which actors are chosen for such role instances

People and hats

- /// At any one time an actor can be acting many different role instances
- /// They have different 'hats' they can wear
- /// At a given moment they have a particular hat on
- /// They can switch hats – they switch roles
- /// They get rid of hats – they finish a role
- /// They acquire new hats – they get given a role

A person is not a hat

- /// Hats don't need wearers to exist
- /// They have their own lives



Identify the types of the roles on our list

- /// Which are pure responsibility?
- /// Which are job titles?
- /// Which come and go, and which are permanent?



A choice to be made

- /// The mapping between pure responsibilities and job titles is a matter of *organisational design*
- /// Whether to model in terms of pure responsibilities or job titles is a *mapping decision*
- /// We decide according to the reason we are modelling: design, definition, improvement, . . .

Roles can be ‘instantiated’

- /// Responsibilities can be created dynamically:
 - /// When a new project is needed, the responsibility for managing the project is created
 - /// When a new clinical trial is needed, the responsibility for running the trial is created
 - /// When a new IT system is to be built, the responsibility for its design is created
 - /// When the people arrive with a reservation, the responsibility for their care is created

Identify some interactions that might occur in our restaurant



/// Which roles take part?

Interactions

- /// Don't think of interactions as just hand-offs or flows
- /// They are collaborative acts
- /// Some add value
- /// They can be simple or complex
- /// They can involve any number of participating roles

Interactions – some general types

- /// A and B discuss
- /// A and B agree
- /// A and B negotiate
- /// A contracts with B
- /// A informs B
- /// A reports to B
- /// A delegates to B
- /// A oversees B
- /// A manages B
- /// A instructs B
- /// A and B carry out an action jointly

A role is a network of threads of

/// actions

/// interactions

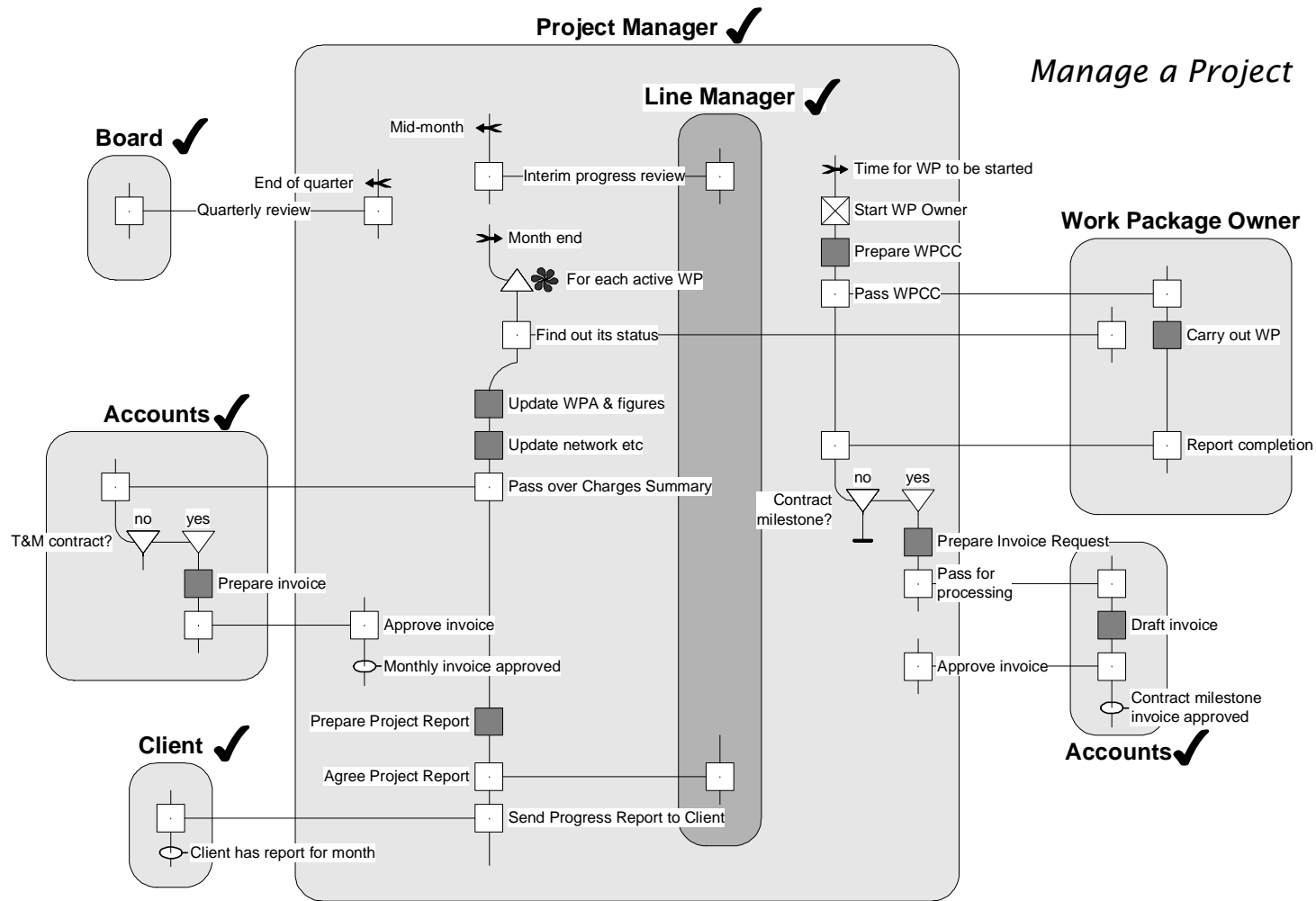
with

/// decisions

/// triggers

/// other structures

pulling them together



A sample *Riva* process model

(Note that any process model, including this one, is drawn for a particular purpose, to answer a specific question)

Concrete vs abstract models

- /// A 'concrete' model shows the *mechanisms* of roles, actions, interactions; eg
 - /// role: *Chief Financial Officer*
 - /// action: *Complete Form 21B*
 - /// interaction: *Send Form 19A*
- /// An 'abstract' model shows their *intent*; eg
 - /// role: *Budget Approving*
 - /// action: *Forecast cashflow for coming year*
 - /// interaction: *Claim expenses*

Using the two model types

- /// Concrete models can be used
 - /// for work instructions
 - /// for understanding an existing process
- /// Abstract models can be used
 - /// for designing a new process
 - /// to identify potential improvements
 - /// to identify technology

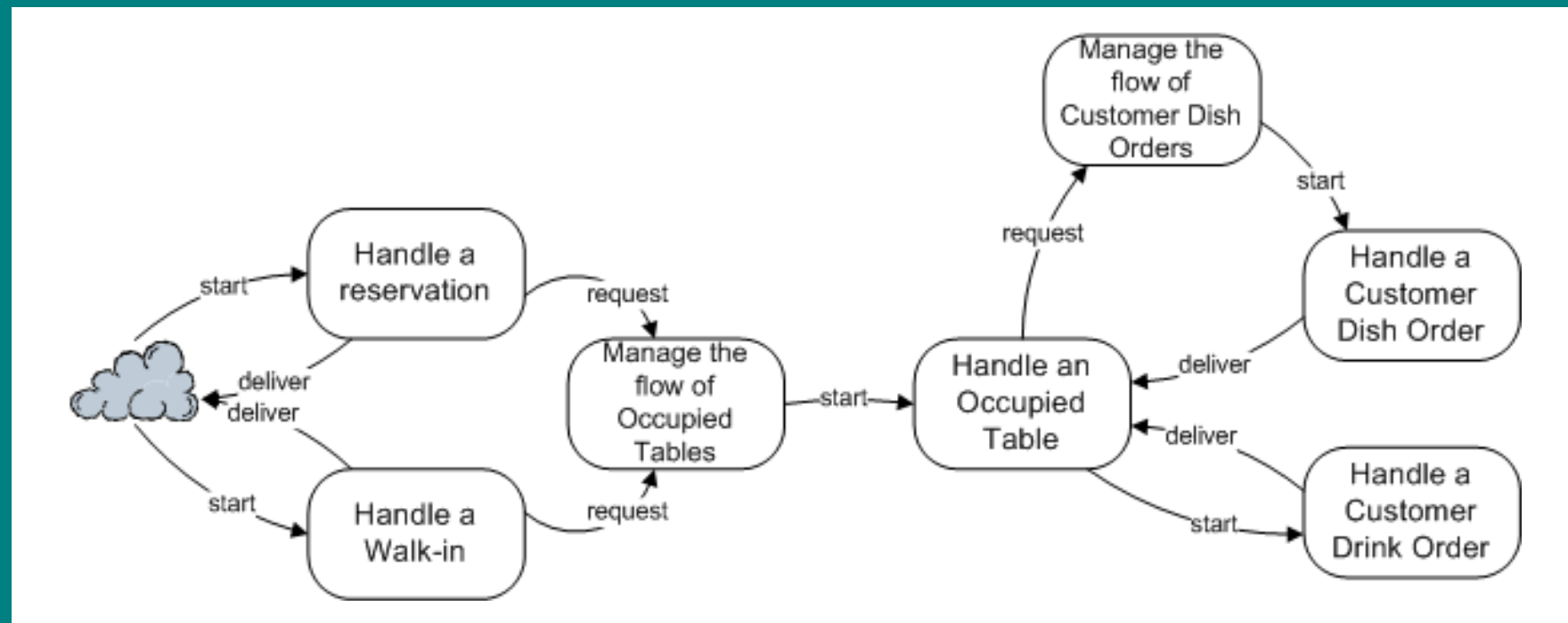
Let's choose a process to
design from our architecture



Which roles take part?

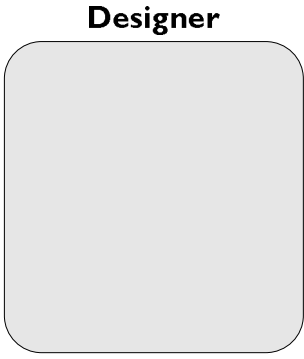


What will those *process interactions* look like when we design our process?

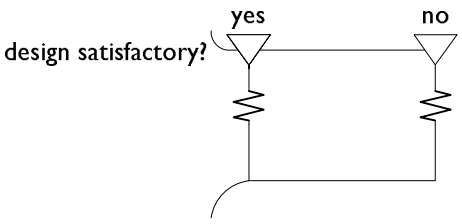


Let's start the design . . .

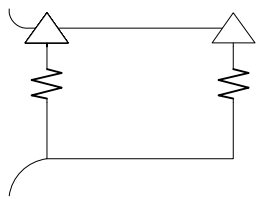
The main RAD notation



A role



Alternative paths depending on the condition: 'case refinement'

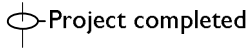


Concurrent paths: 'part refinement'

A state



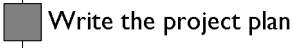
A state description



A trigger



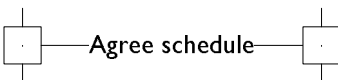
An action



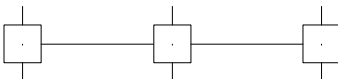
Start another role



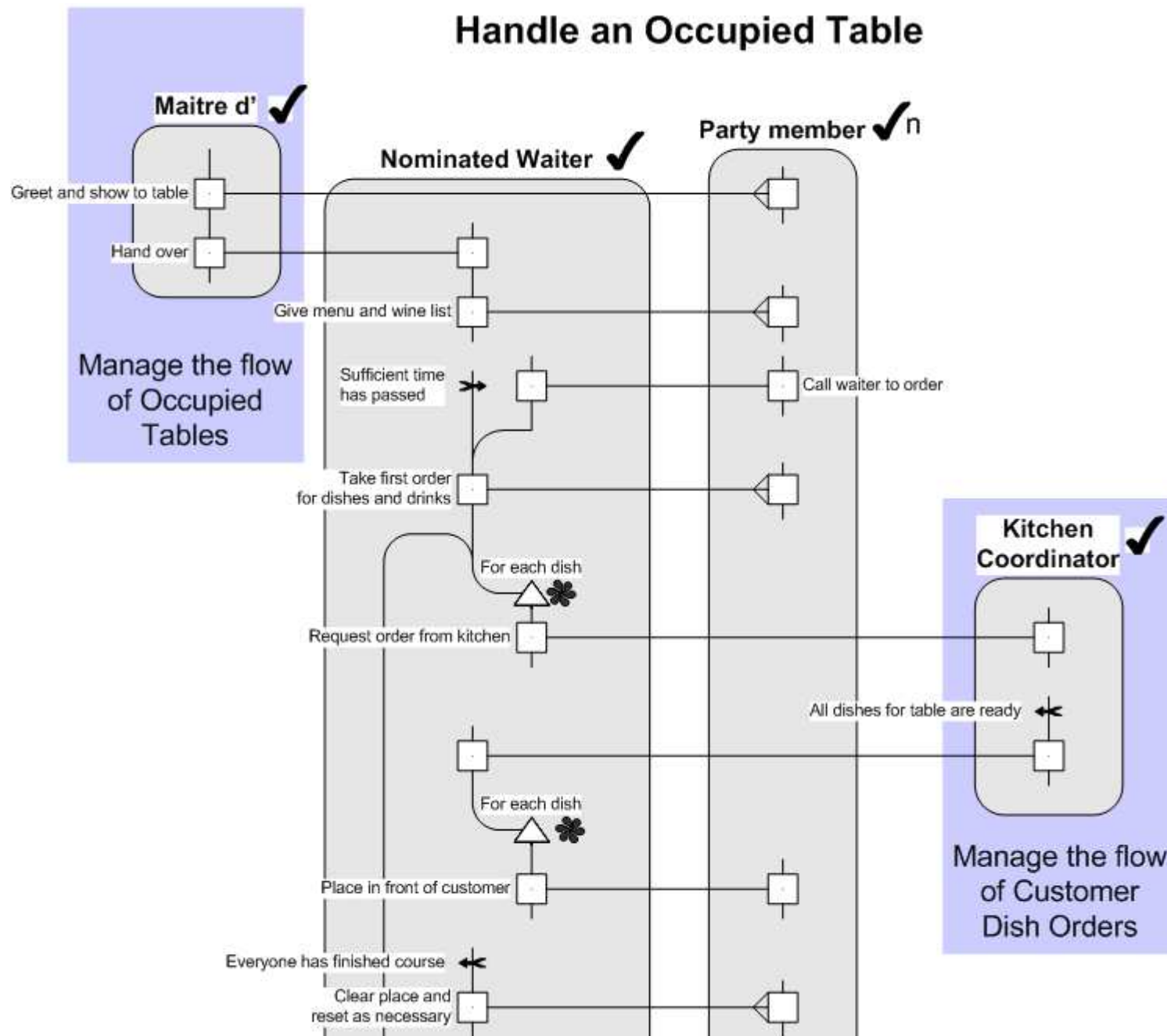
An interaction between two roles



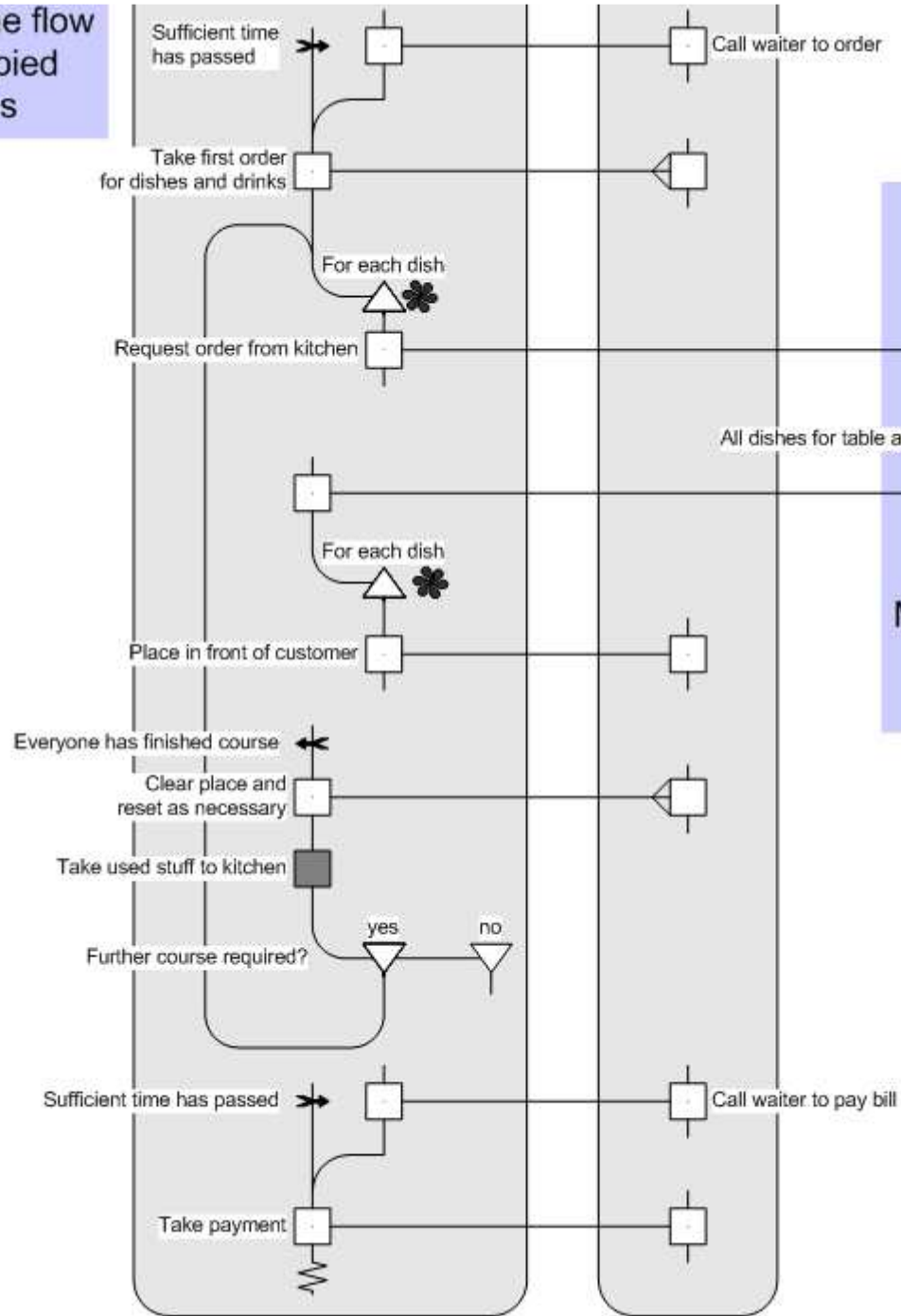
An interaction between three roles



Handle an Occupied Table



Manage the flow of Occupied Tables



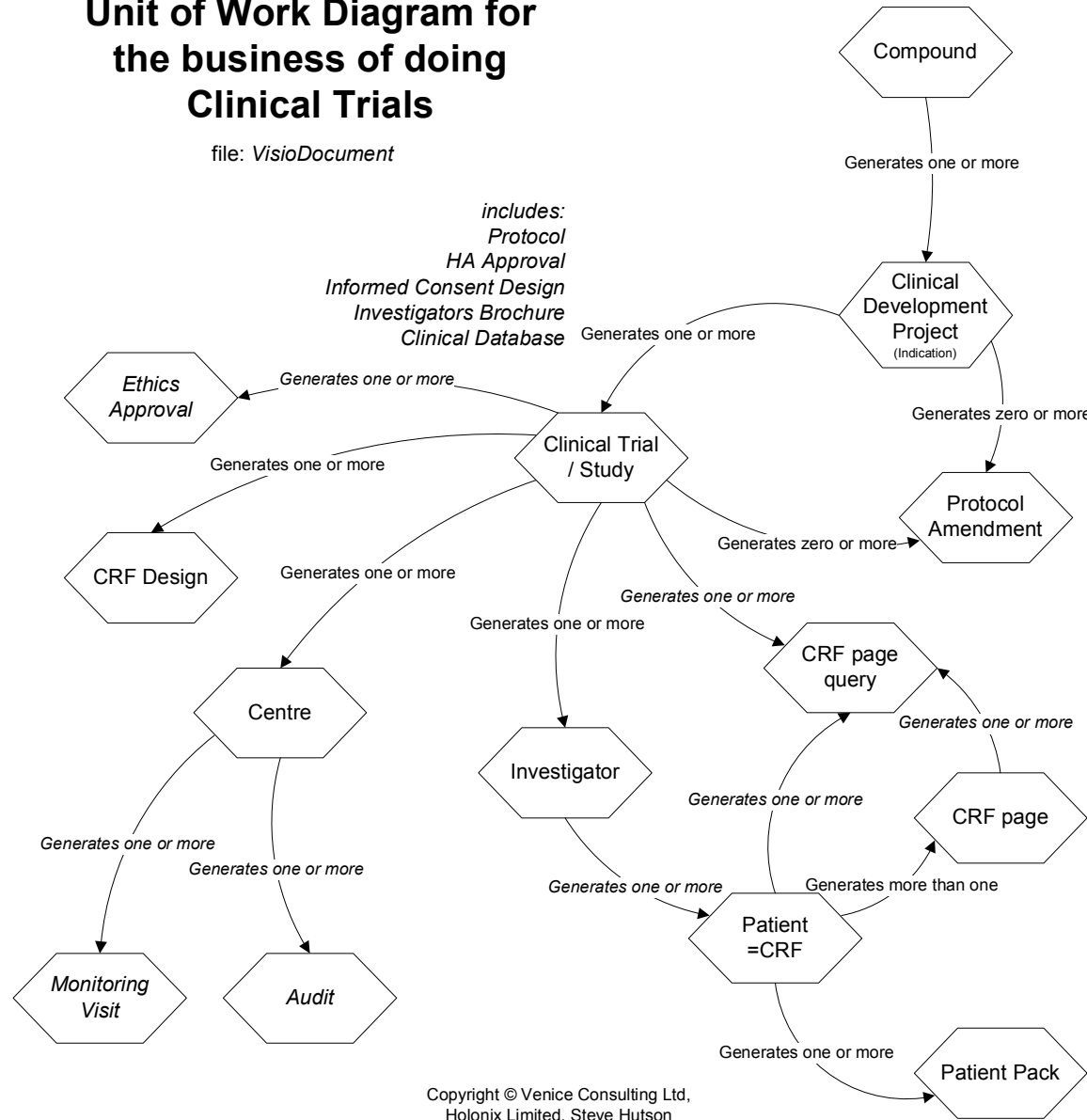
Kitchen Coordinator ✓

Manage the flow of Customer Dish Orders

An example from real life

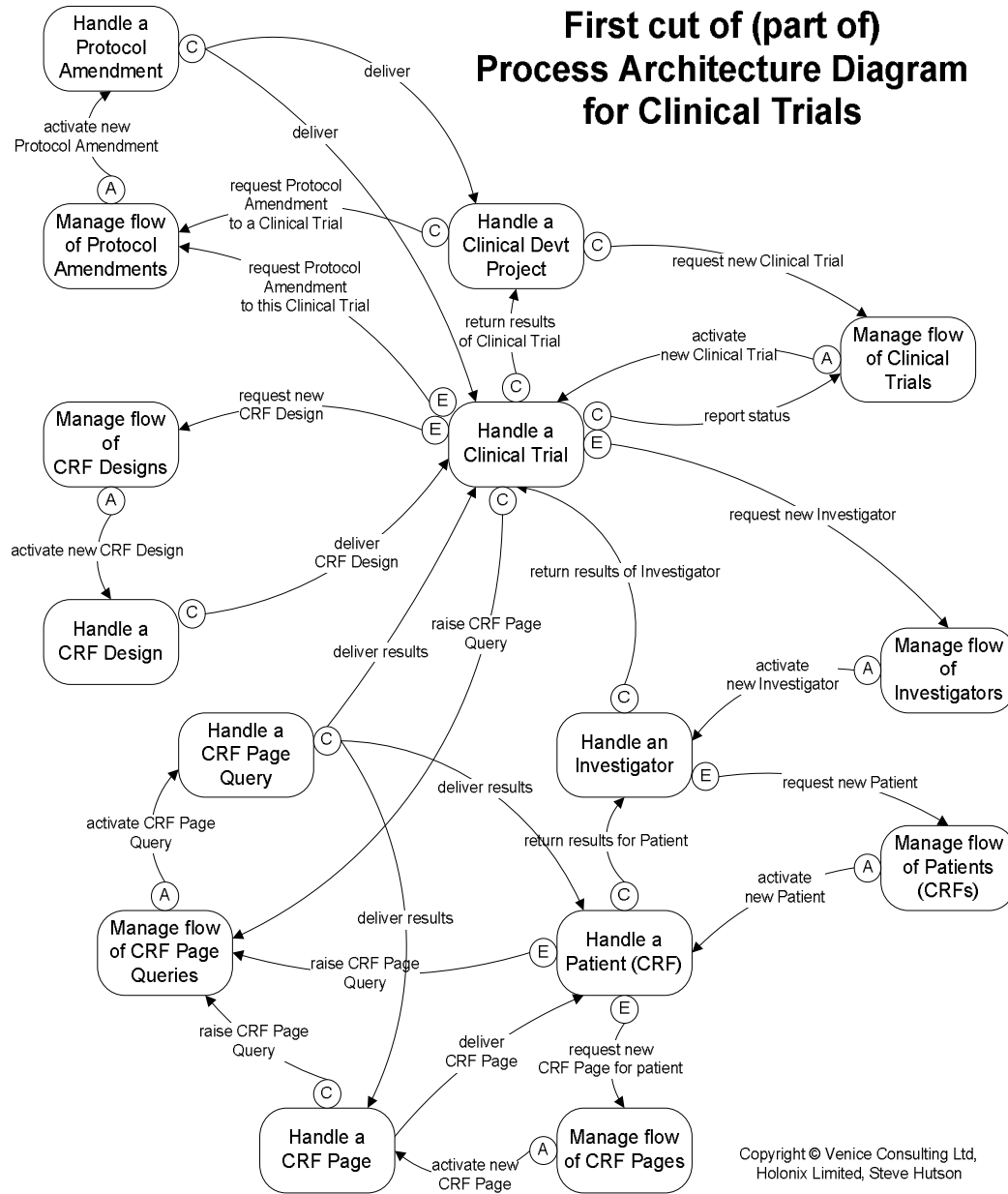
Unit of Work Diagram for the business of doing Clinical Trials

file: VisioDocument



Clinical trials Unit of Work Diagram

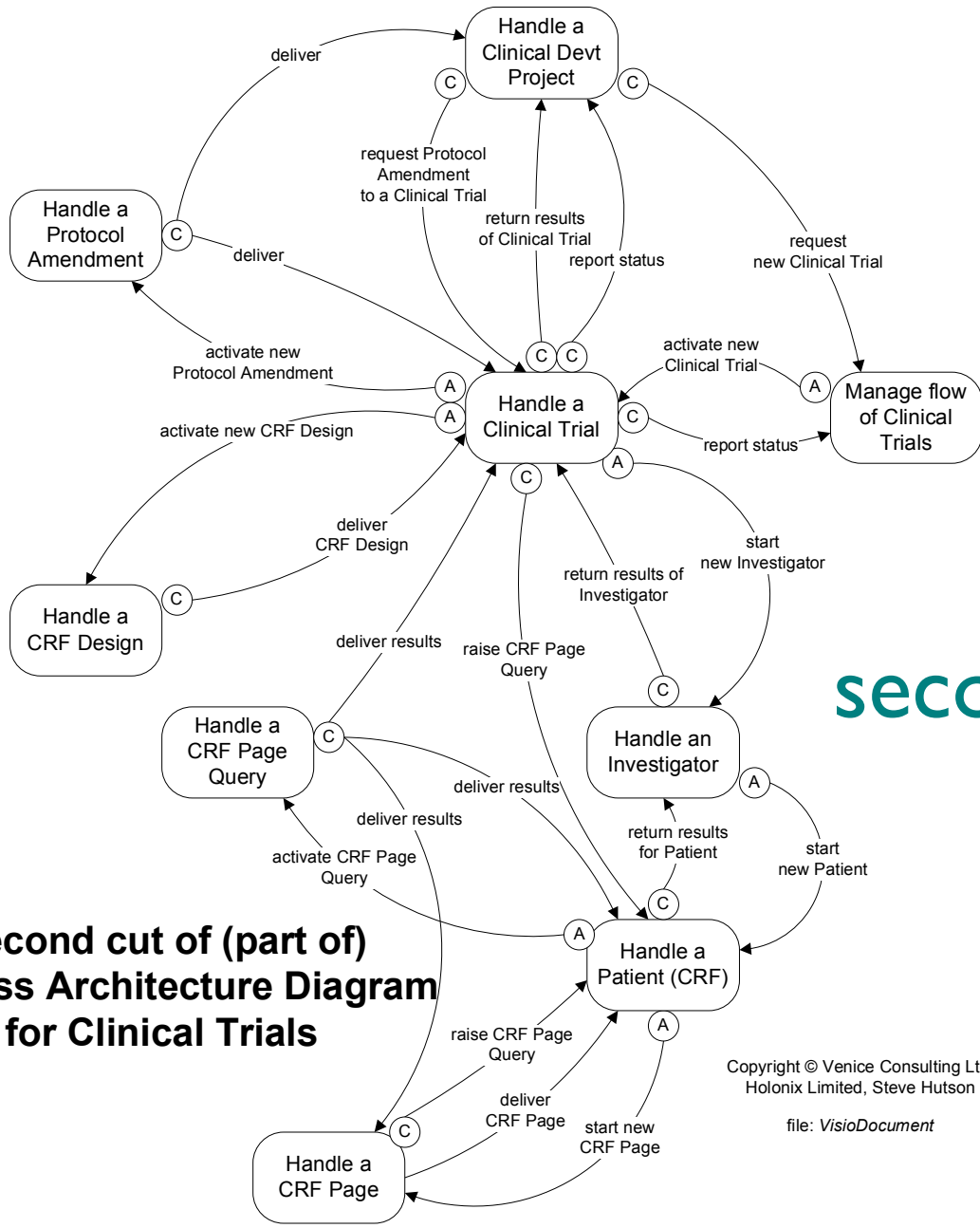
First cut of (part of) Process Architecture Diagram for Clinical Trials



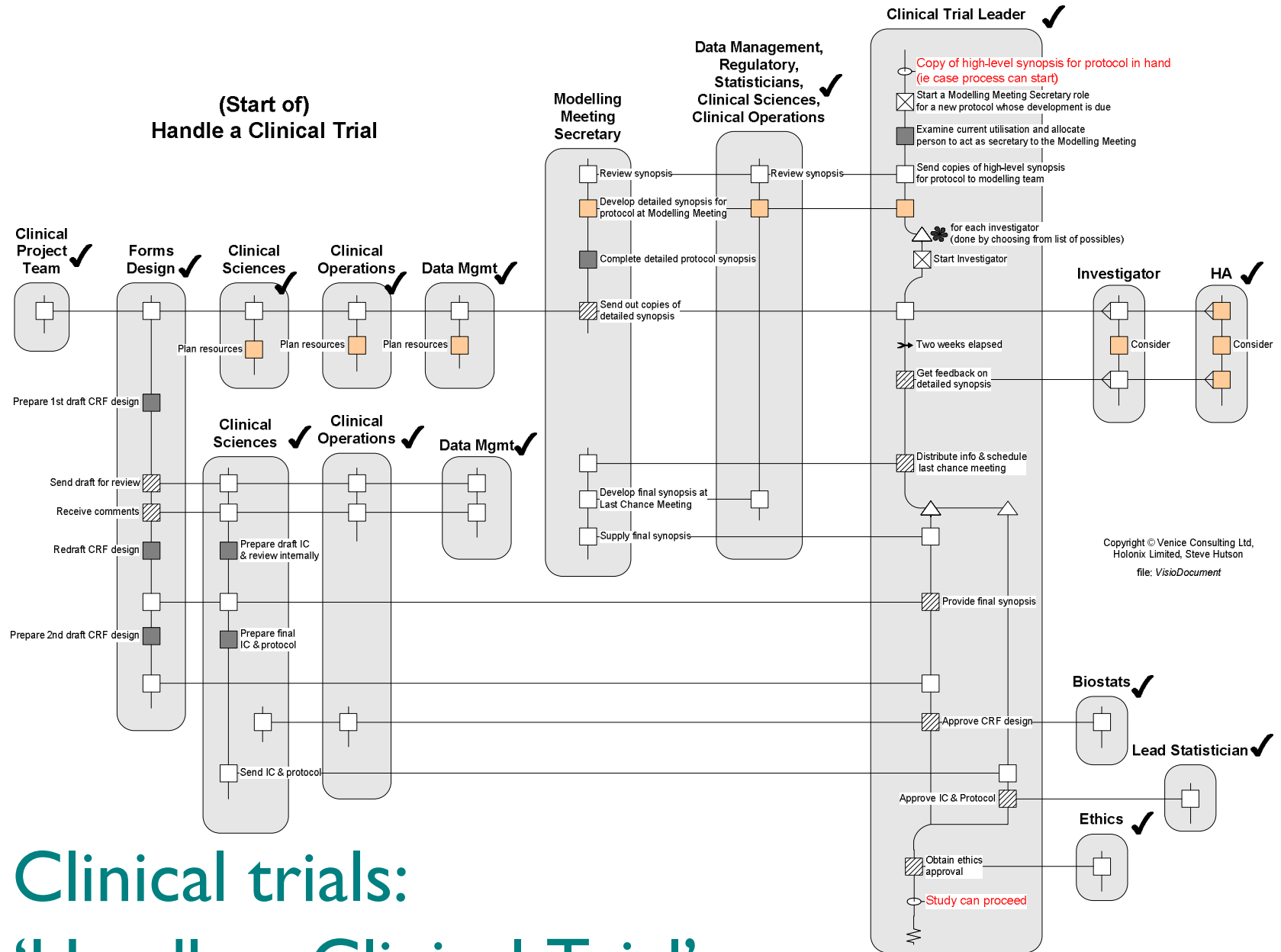
Clinical trials
first-cut process
architecture

Clinical trials second-cut process architecture

**Second cut of (part of)
Process Architecture Diagram
for Clinical Trials**



Copyright © Venice Consulting Ltd,
Holonix Limited, Steve Hutson
file: *VisioDocument*



Clinical trials: 'Handle a Clinical Trial'

Time to summarise



The secret of modelling concurrency: *instantiation*

- /// At any moment we see a set of *instances*
 - /// of processes
 - /// of roles within process instances
 - /// of threads within role instances
- /// That set is constantly changing
- /// Instantiation in **Riva** captures concurrency and the dynamics of the organisation at many levels

The secret of modelling collaboration: *interaction*

- /// We identify and model interactions
 - /// between processes
 - /// between roles
- /// Interactions in ***Riva*** capture collaboration in the organisation at both levels

Riva works with networks

- /// Organisational activity is a network of interacting processes
- /// A process is a network of interacting roles
- /// A role is a network of connected actions, decisions and interactions

- /// We allow no artificial or arbitrary hierarchies!

The benefits of using *Riva*

By concentrating on the right things, it is

- /// business-focused
- /// thorough
- /// quick
- /// productive

*This presentation can be loaded for personal use from
www.veniceconsulting.co.uk/2016GartnerBTPM.pdf
or from*

tinyurl.com/zqxvn56

Venice Consulting



The *Riva* method is described in

Business Process Management A Rigorous Approach

published by BCS and Meghan-Kiffer (North America)



Venice Consulting

