



DIT045 H17 Requirements and User Experience

Requirements Modeling

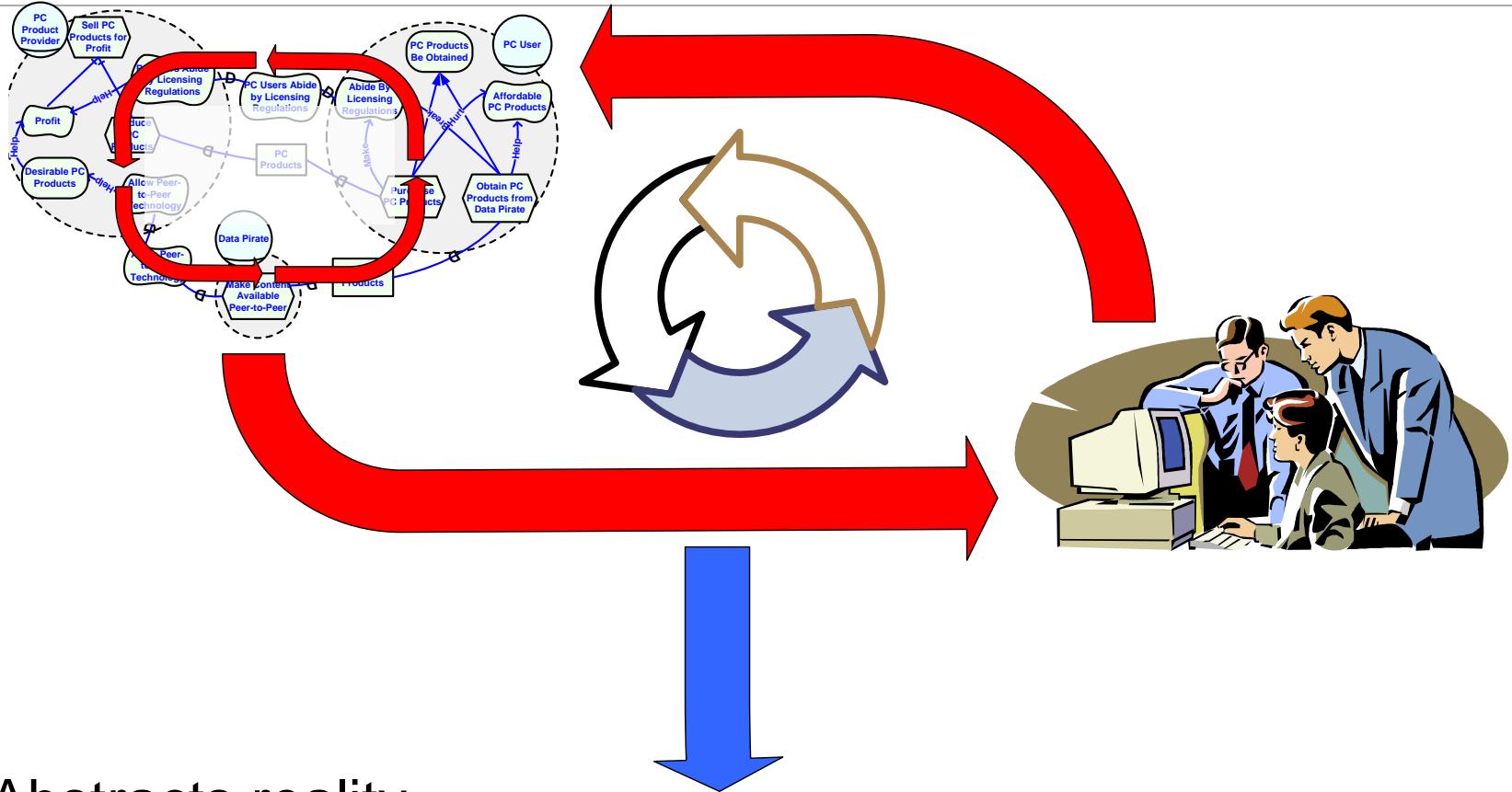
Jennifer Horkoff

Email: jenho@chalmers.se

Why Modeling?

- Visual summary
- Forces you to think of specific concepts and relationships
- Forces you to think in different ways, depending on the type of model
 - Context model: think of system scope and actors
 - Goal model: actors, goals, dependencies, qualities....
 - Use Case model: actors, functions

Why Use Requirements Models?



- Abstracts reality
- Facilitates communication and understanding
- Reveals gaps in knowledge
- Promotes domain analysis, “what if...?”
- Helps elicit requirements

Requirements Modeling

- Many types of requirements models, we pick 3 types
- Most of UML can be used to model interactions between the world and the system
 - Class diagram: entities in the world
 - Activity diagram: user activity flows
 - Sequence diagram: user sequence of actions
 - State diagrams: states of entities in the world
- We won't do (very much) UML modeling in this course
 - You'll see in other courses later
 - (Just remember you can also use these models for requirements)
 - Use cases are part of UML
 - Context diagrams are not (they are related to Data Flow Diagrams)
 - Goal models are not (they are more specific to RE)

Tools

- Microsoft Visio (not free, academic license)
- PowerPoint/Word
- Draw.io (free, no templates)
- Creatively (4 free, diagrams public, templates)
- Lucidchart (free for some time, templates?)
- Many more....

Context Diagrams

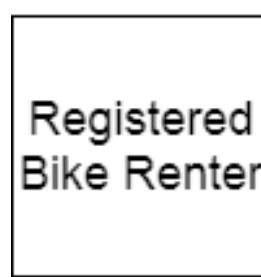
- Two main components:
 - Entities (actors):
 - One in the middle representing the system
 - Others around the outside representing actors providing input or receiving output from the system
 - Relationships:
 - Show the inputs and outputs to and from the system
- Optional helpful component:
 - Scoping tiers via concentric circles (examples later)
- Same as a level 1 data flow diagram
 - You will learn about this with databases next semester

Context Entities

- System actor



- Other actors



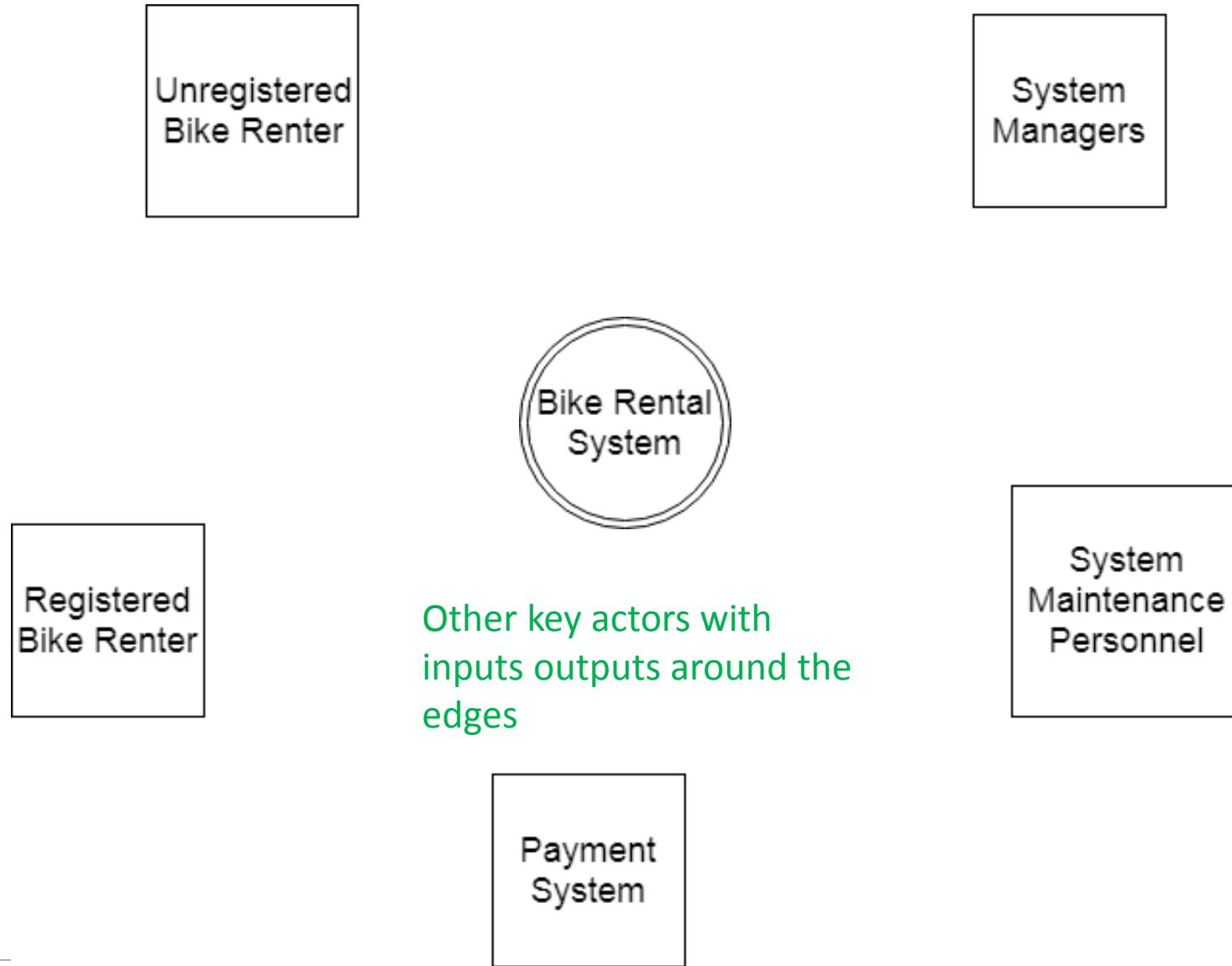
- Note: can use different shapes, just provide a legend
 - Tell us what each shape is supposed to be

Context Example: Bike Rental

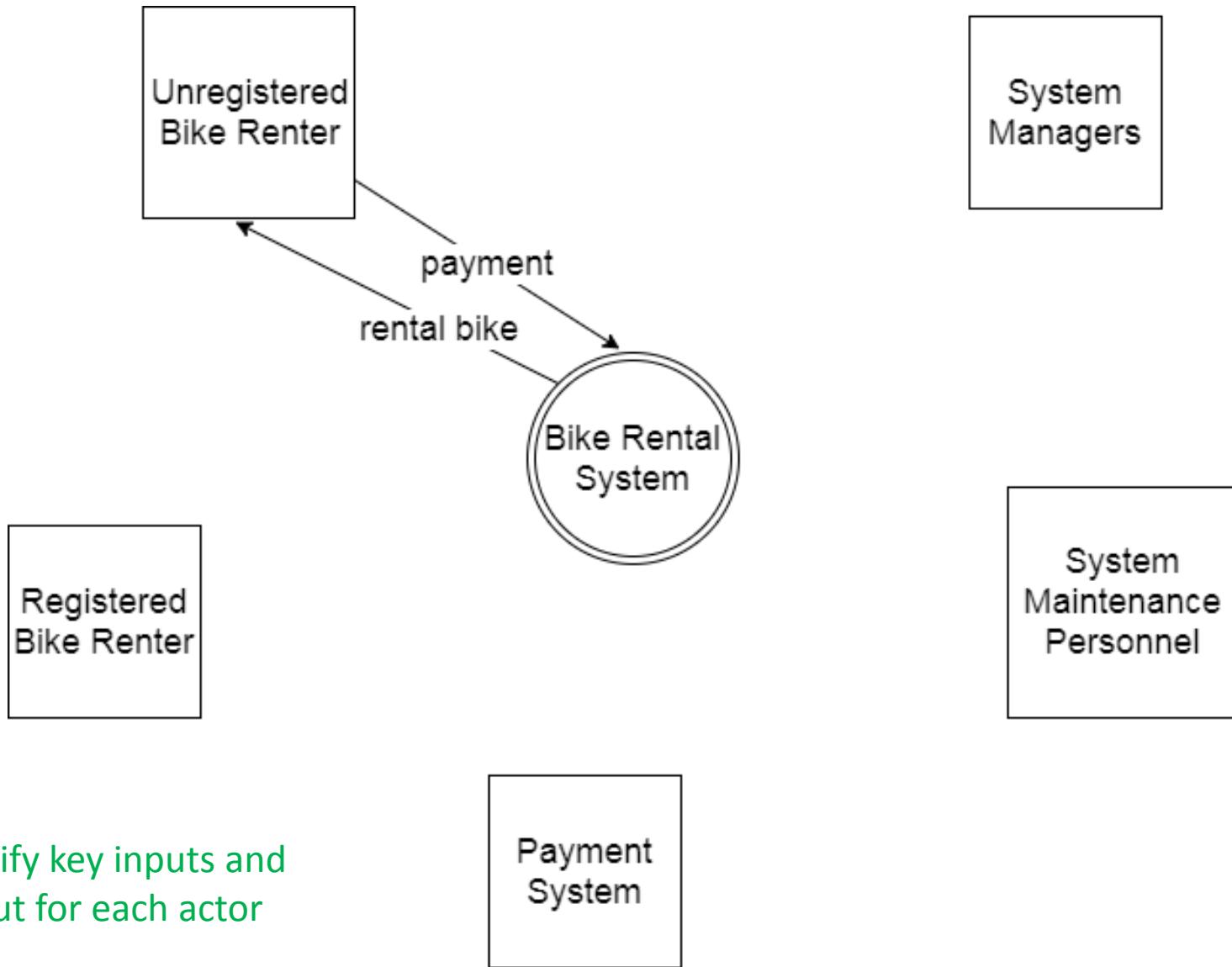
System actor in the middle



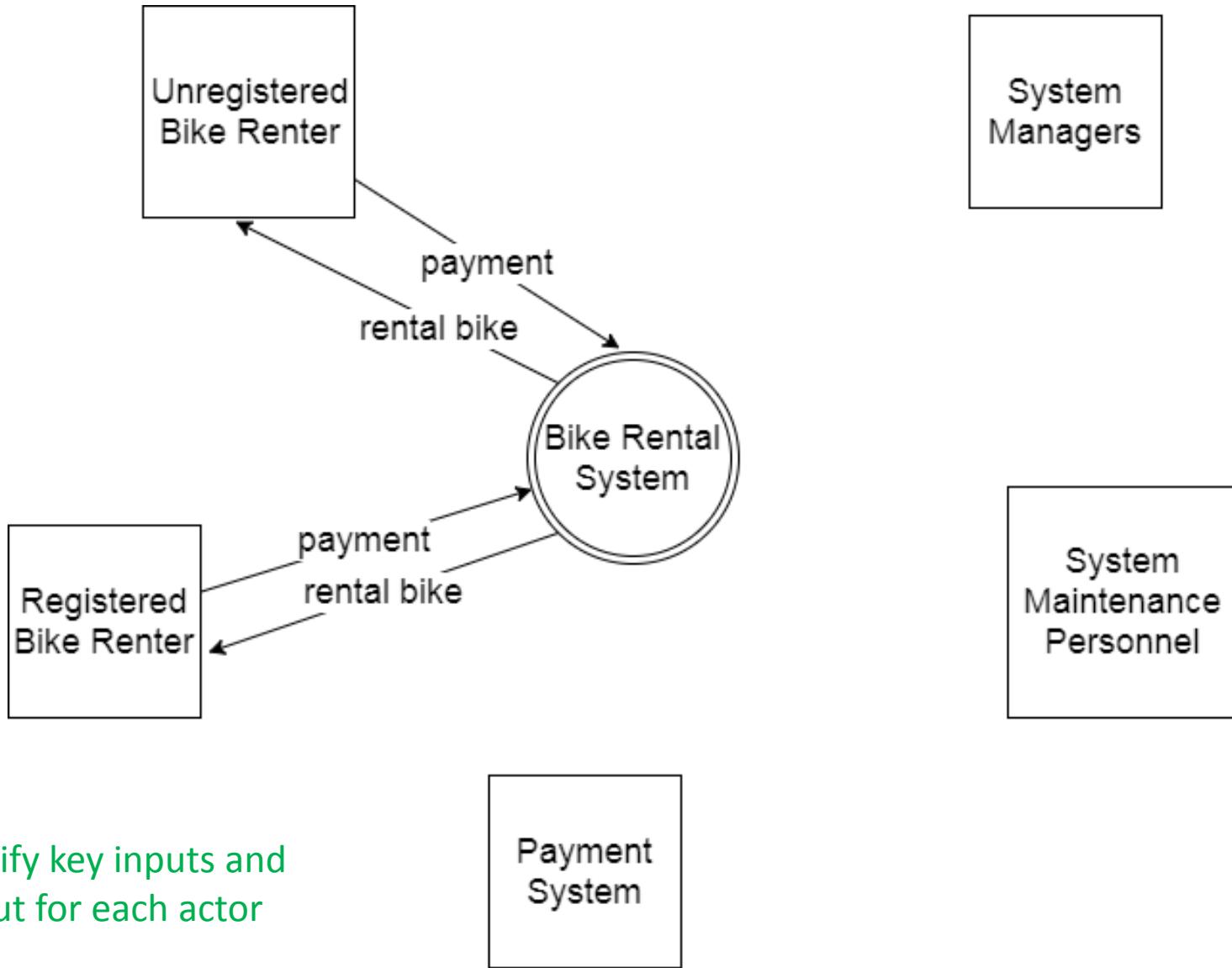
Context Example: Bike Rental



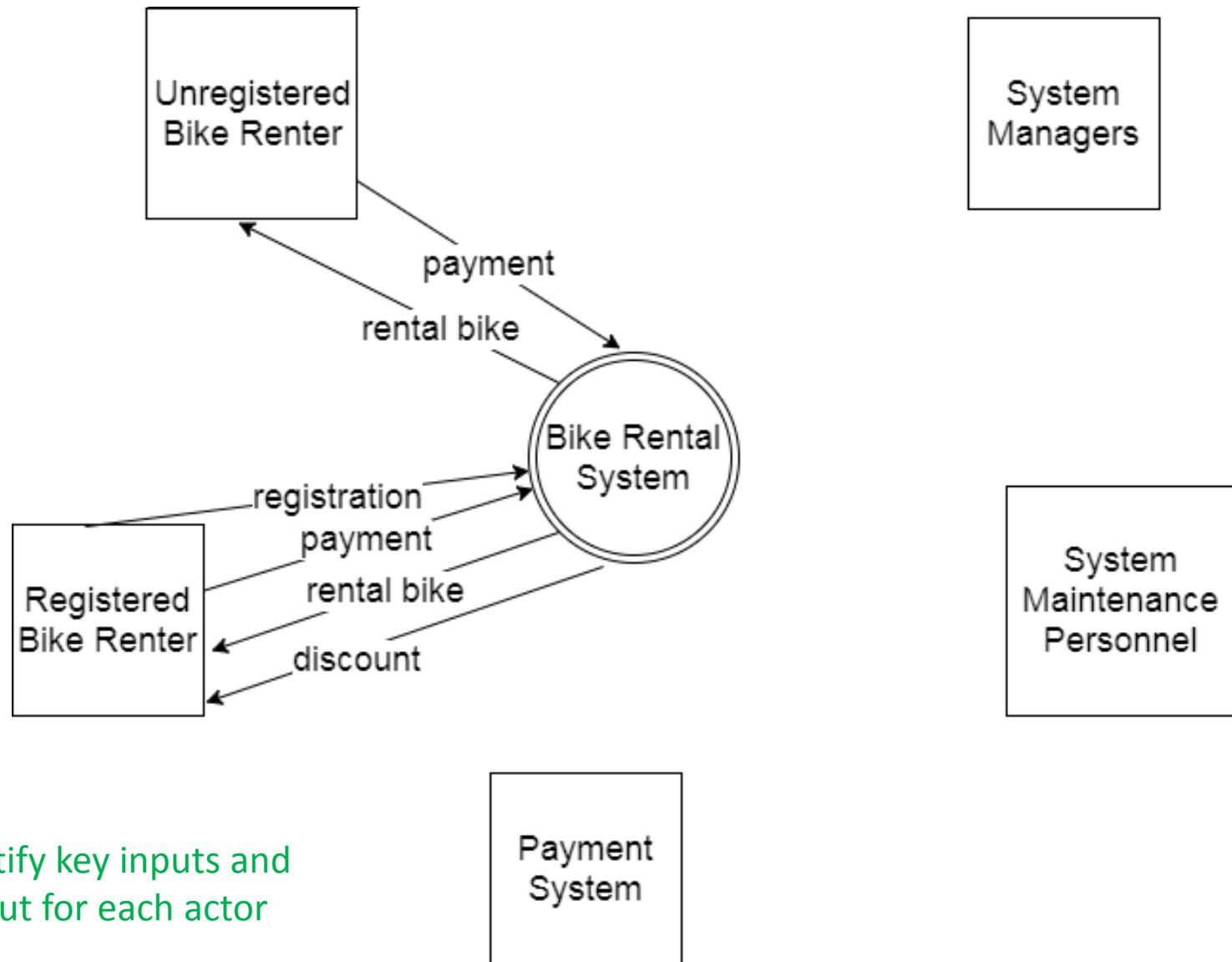
Context Example: Bike Rental



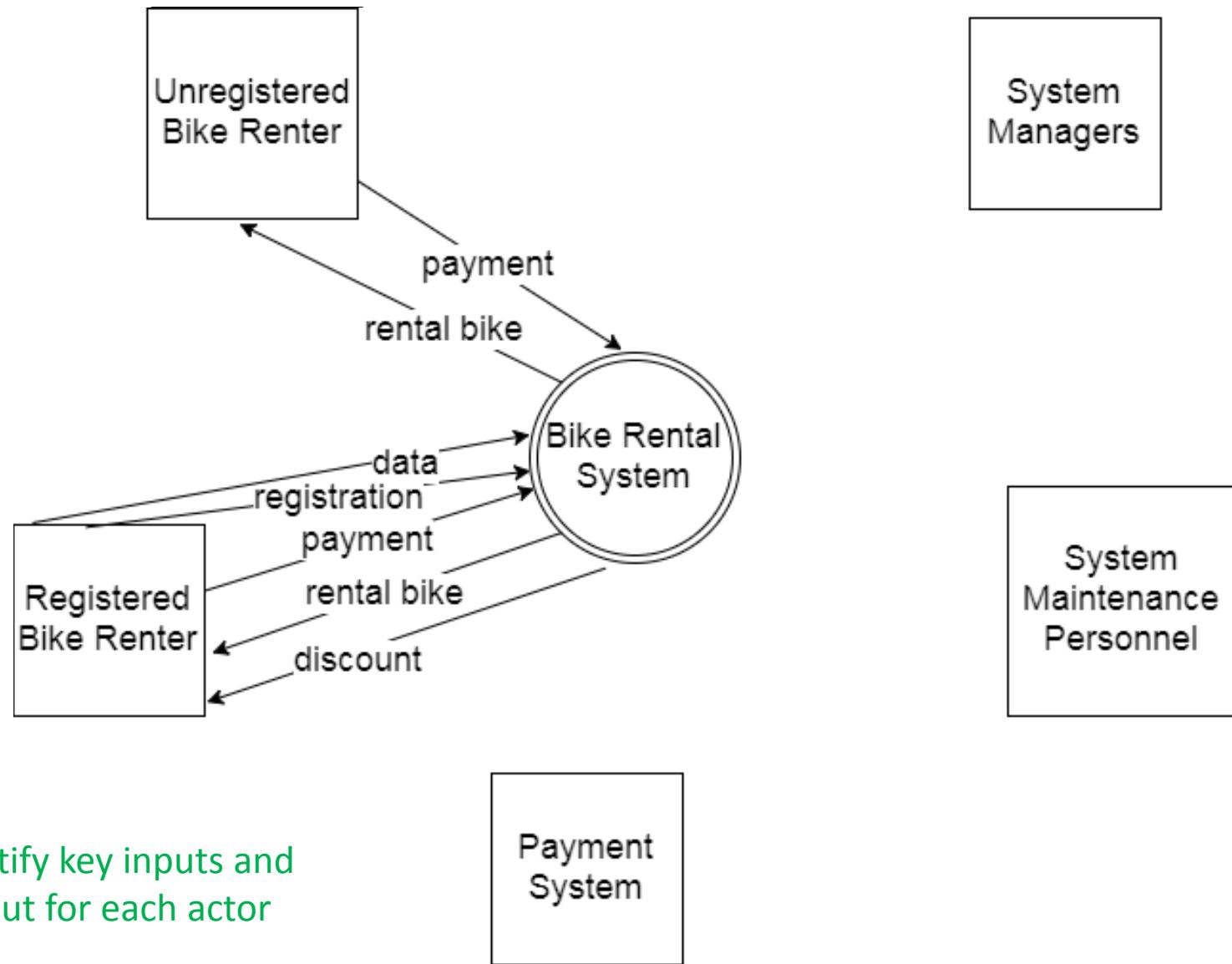
Context Example: Bike Rental



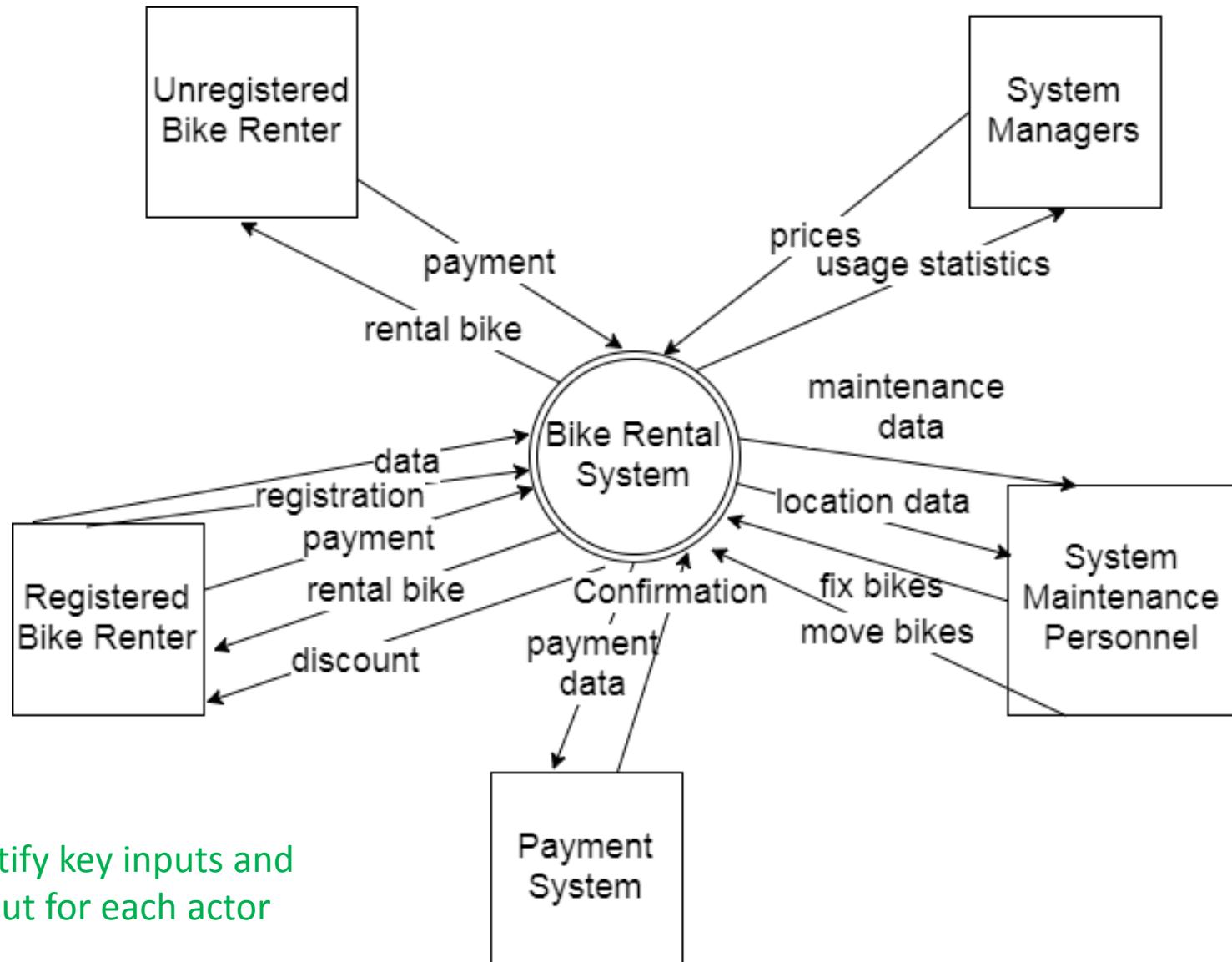
Context Example: Bike Rental



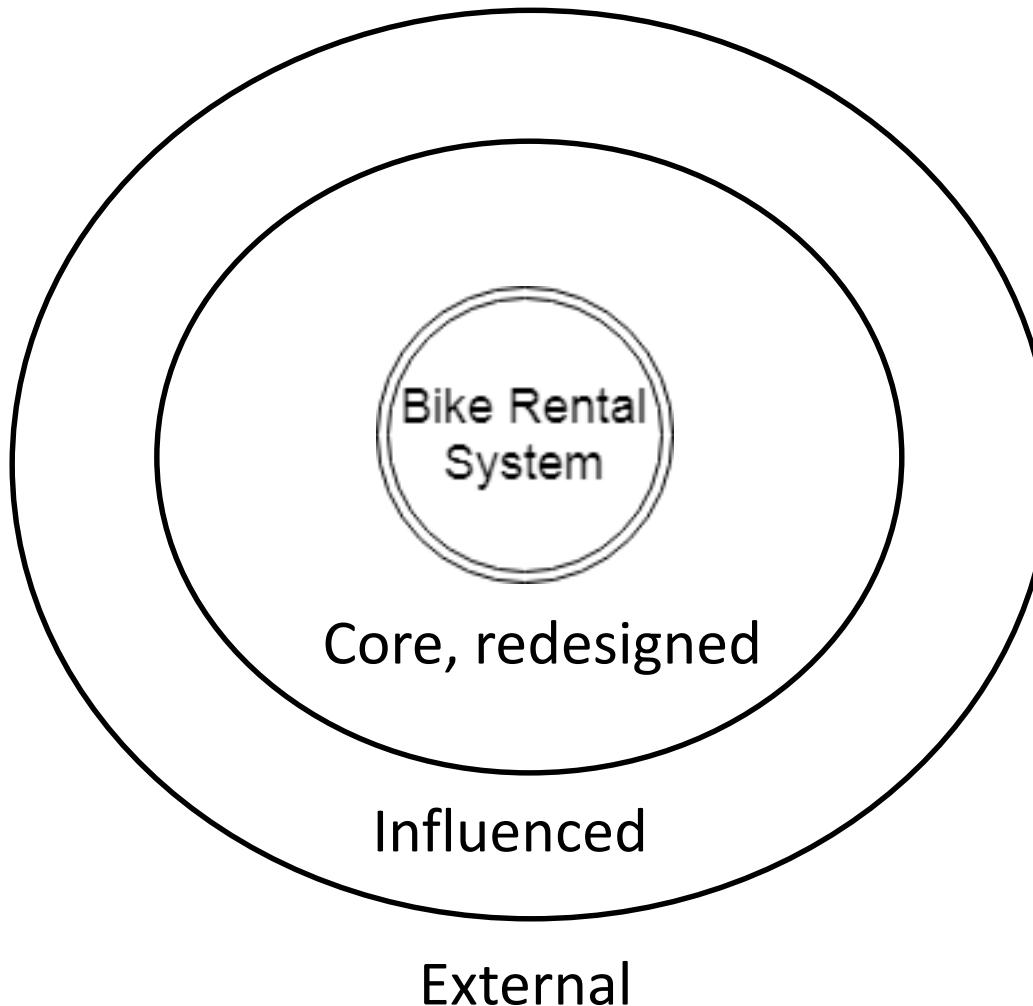
Context Example: Bike Rental



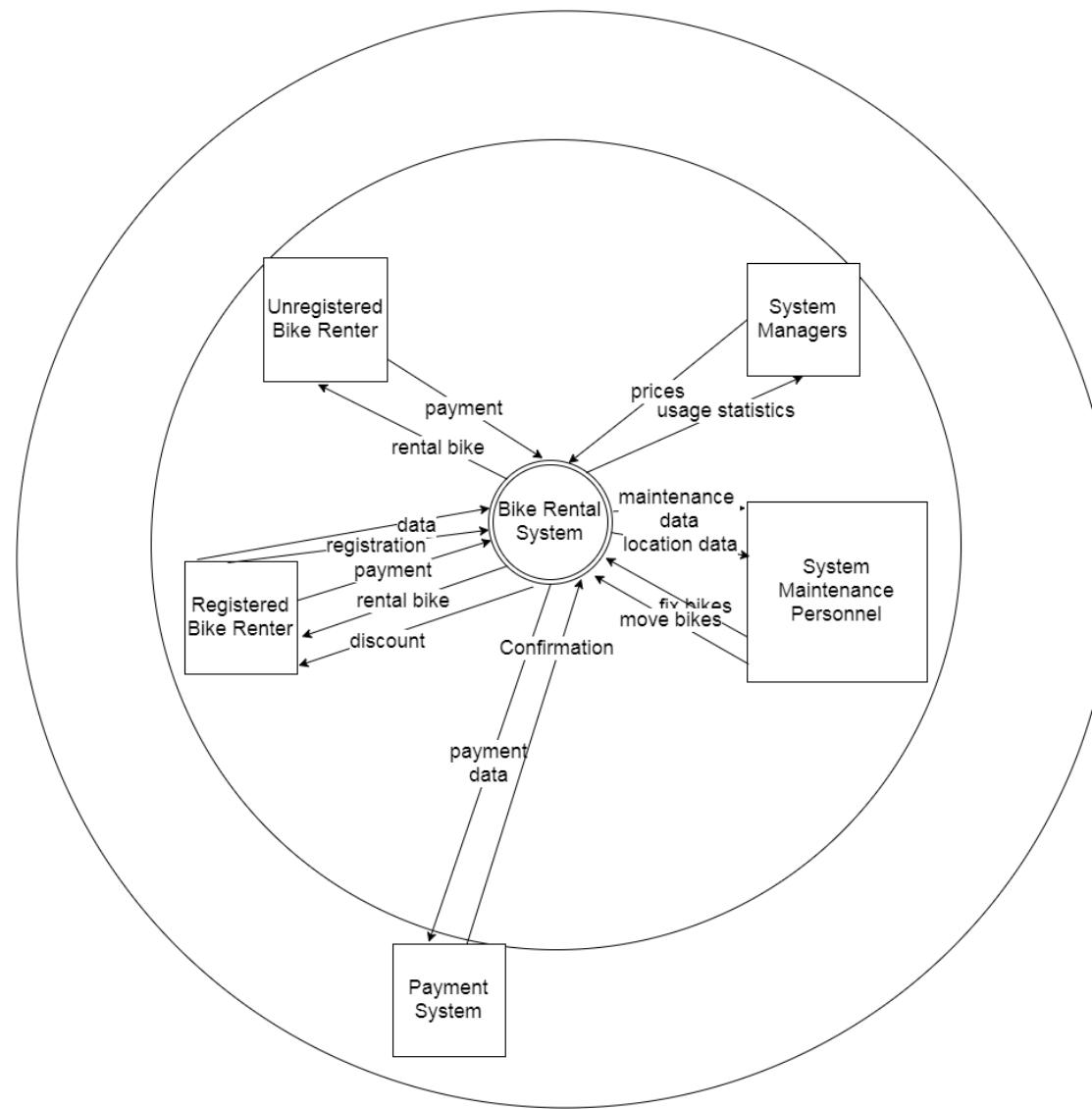
Context Example: Bike Rental



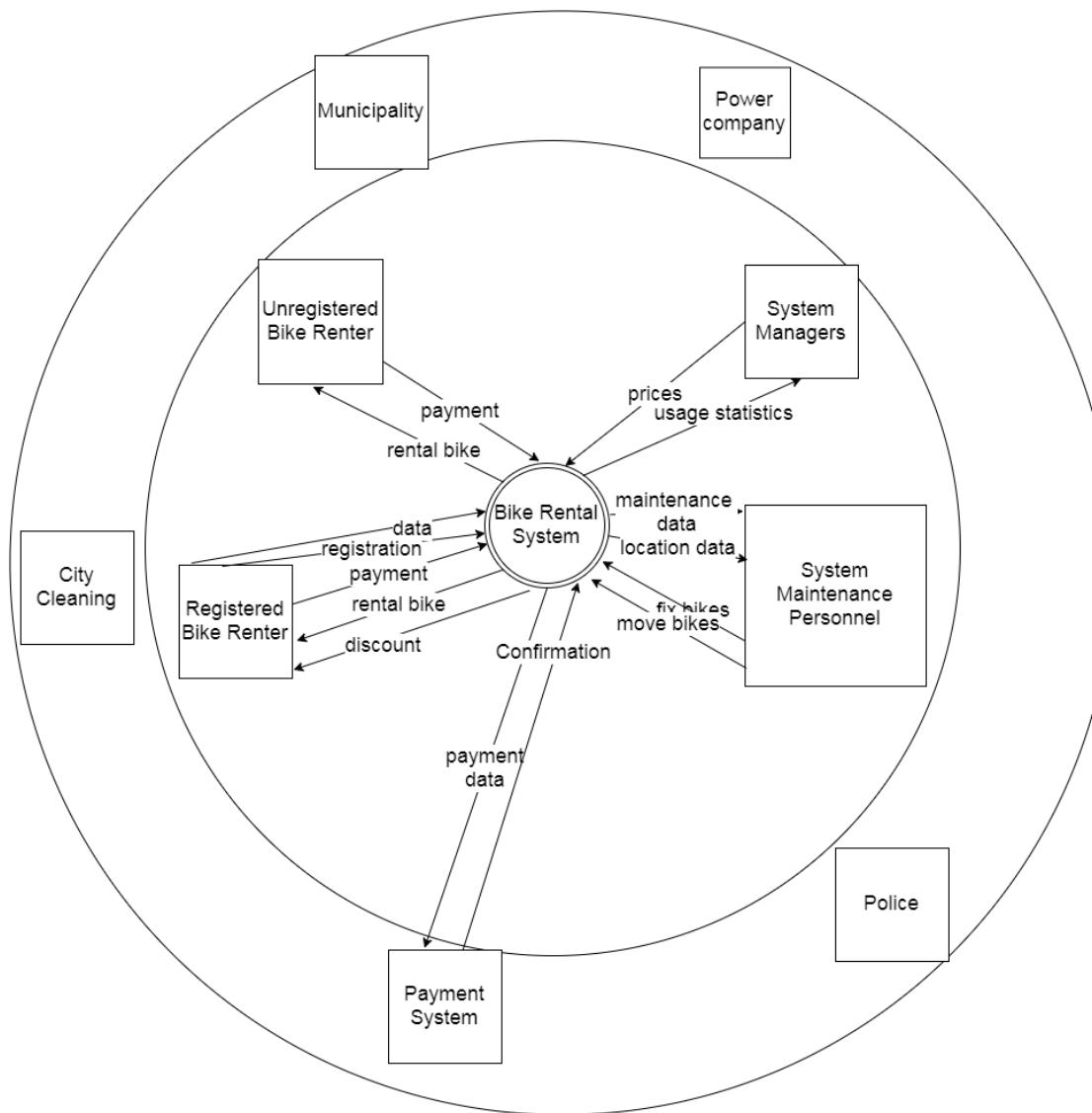
Scoping



Bike Rental

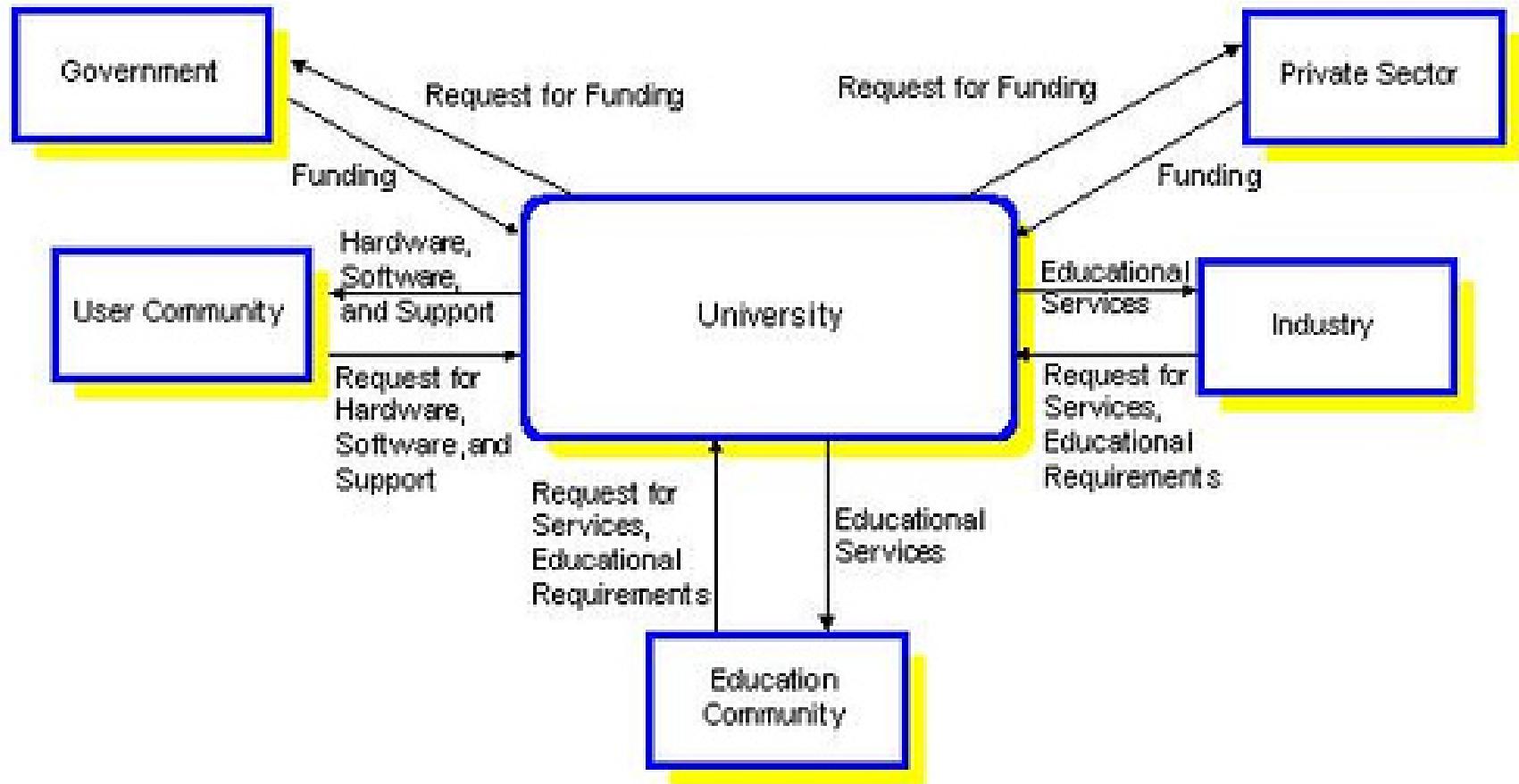


Bike Rental



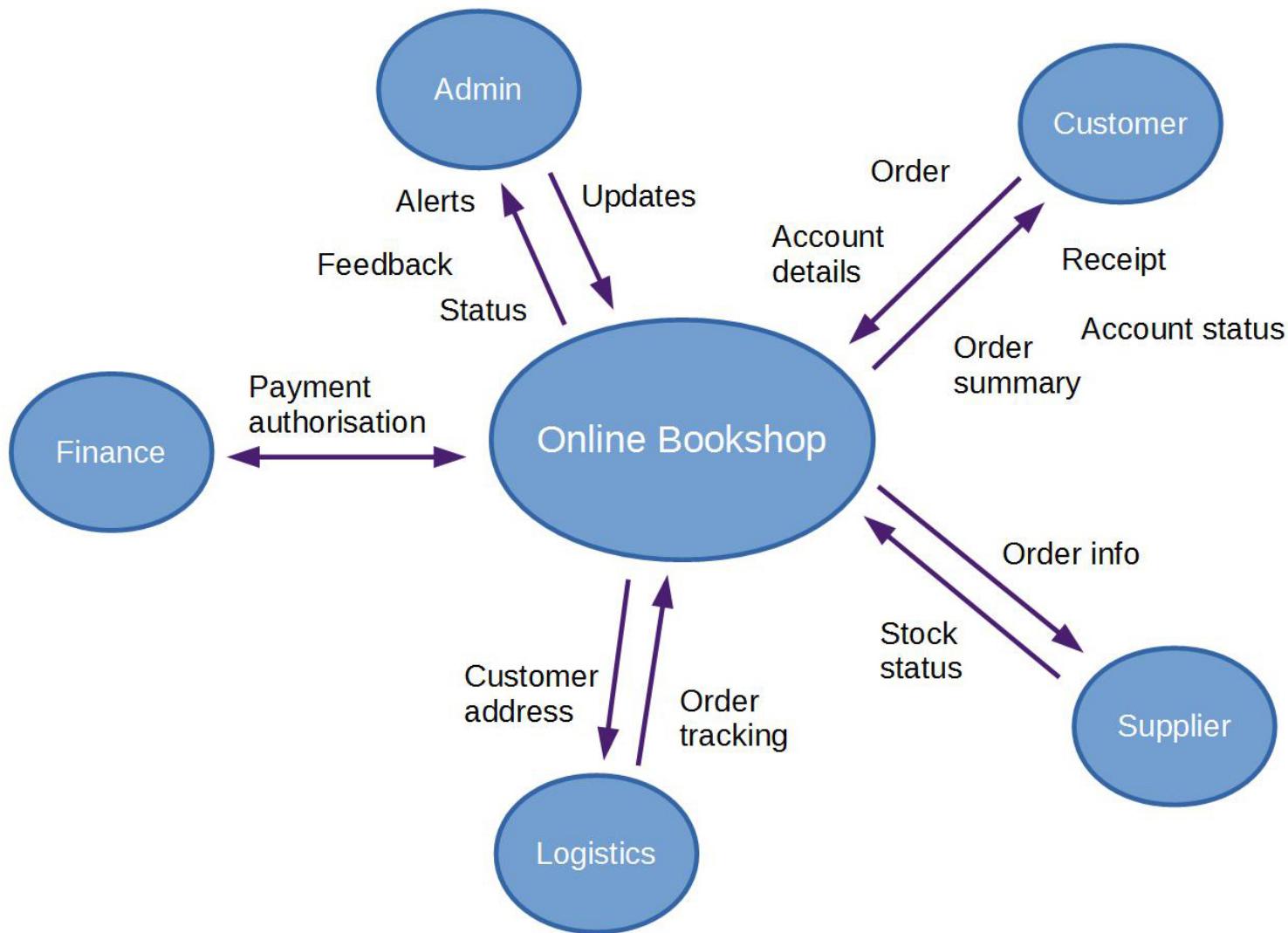
Context Diagram Example

BUSINESS CONTEXT DIAGRAM SAMPLE



<http://it.toolbox.com/blogs/enterprise-solutions/context-diagrams-at-the-enterprise-level-14345>

Context Diagram Example



<https://krakenknowledge.wordpress.com/2014/10/14/design-workshop-week-3-system-context-and-component-diagrams/>

Use Case Diagrams



Use Case Diagrams

- Part of UML (Unified Modeling Framework)
 - Well-known modeling standard
- A visual (model) part and a longer text part
 - We'll start with the model part in A1
- Actors (again)
 - Stakeholders, anyone who uses the system to do anything
- Use cases
 - “**Use cases.** A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.”
 - “Use Case: A use case is all the ways of using a system to achieve a particular goal for a particular user. Taken together the set of all the use cases gives you all of the useful ways to use the system, and illustrates the value that it will provide.” Jacobson, 2011

<http://www.agilemodeling.com/artifacts/useCaseDiagram.htm>

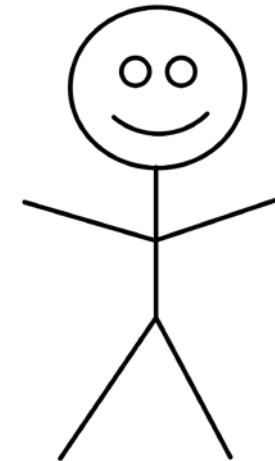
Use Case Diagrams

- **Use Cases**
 - Capture requirements
 - Usually at a higher level than text requirements – shorter and more abstract
 - Are usually broken down into a sequence of actions
 - Have a long form template version – a scenario
 - Will cover scenarios and the longer form of use cases in a later lecture
- **Associations**
 - Which actors are involved in which use cases
- **System boundary**
 - Use cases are shown as part of a system

<http://www.agilemodeling.com/artifacts/useCaseDiagram.htm>

Bike Rental Again

- Actors (can be same as in context diagram)
 - Users (registered user, non-registered user)
 - Maintenance personnel
 - Managers
 - Payment system
 - ... anyone else who interacts with system

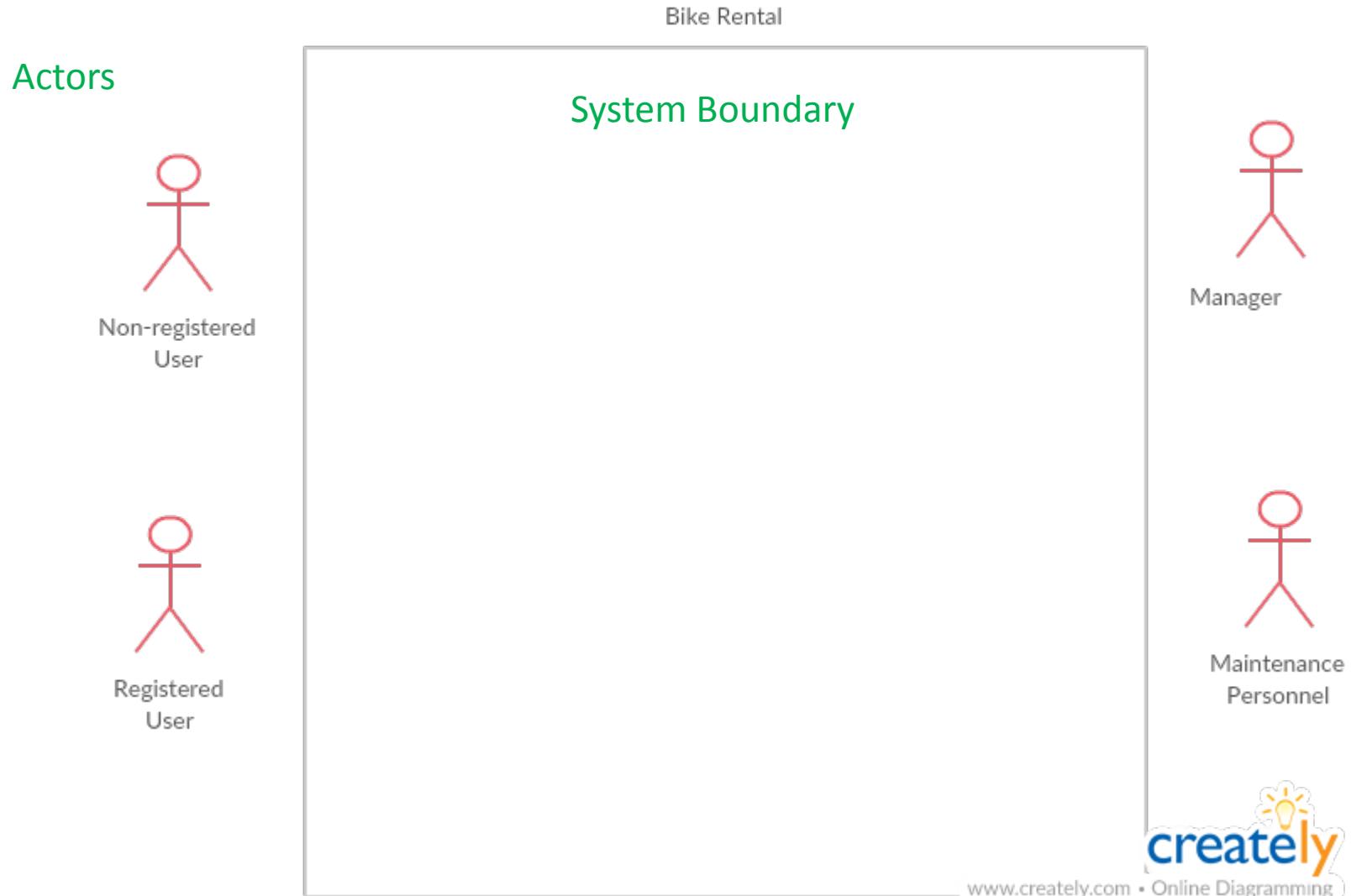


- Use cases (should cover main functionality)

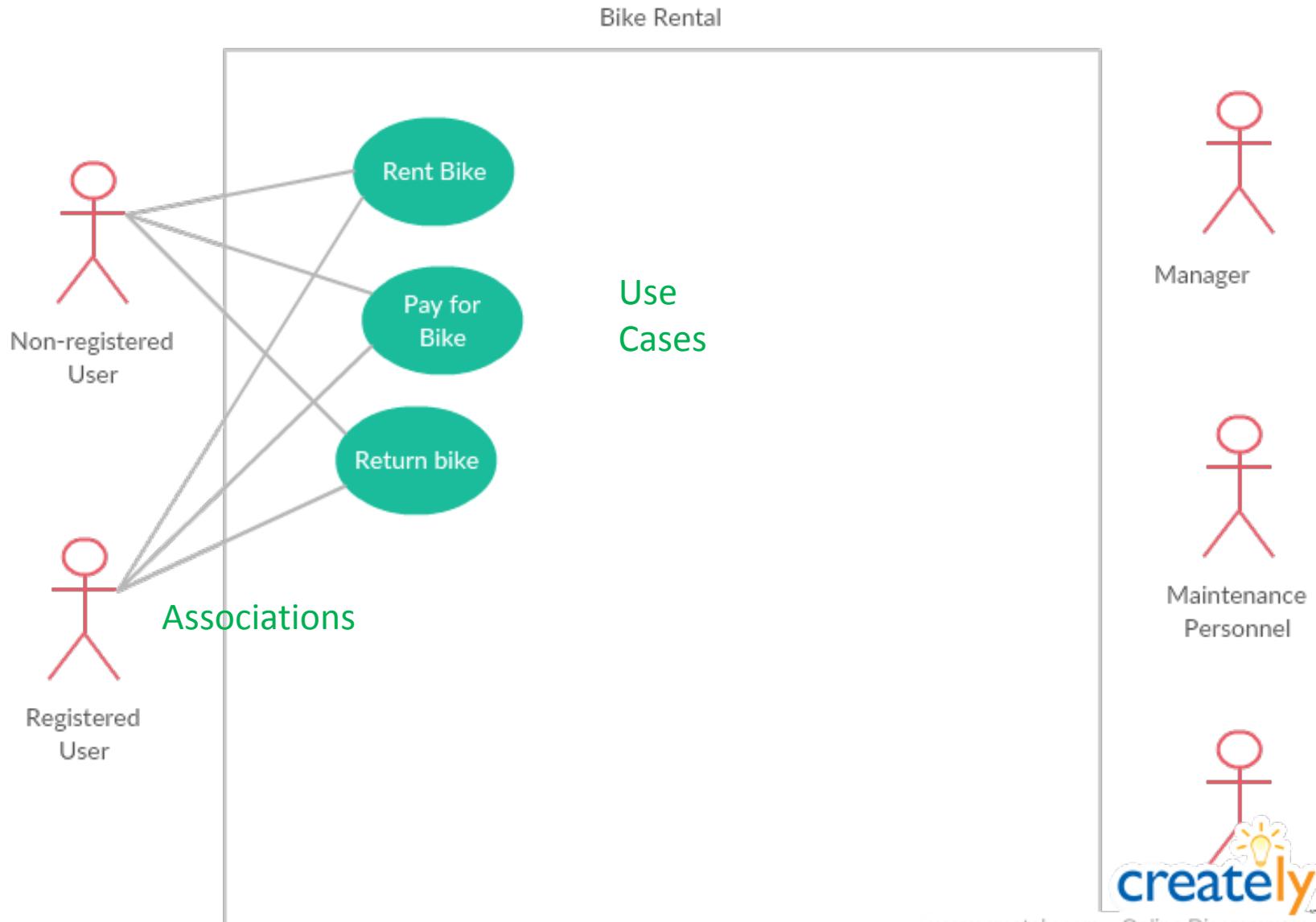
Rent bike	Get maintenance info
Pay for bike	Get bike location info
Return bike	Move bikes
Enter registration info	Get bike usage data
Get discount	

- Who is involved in these use cases?

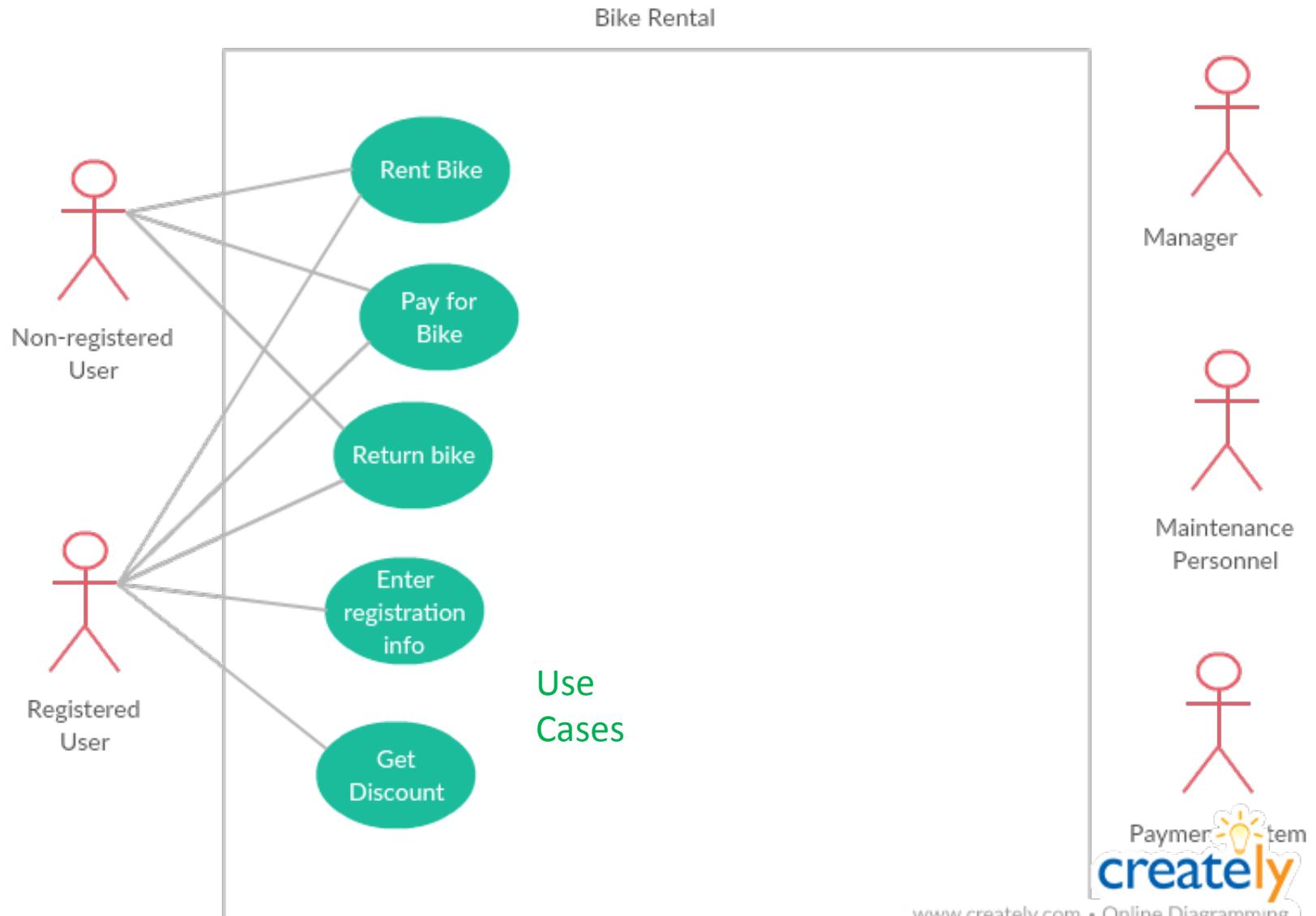
Bike Rental



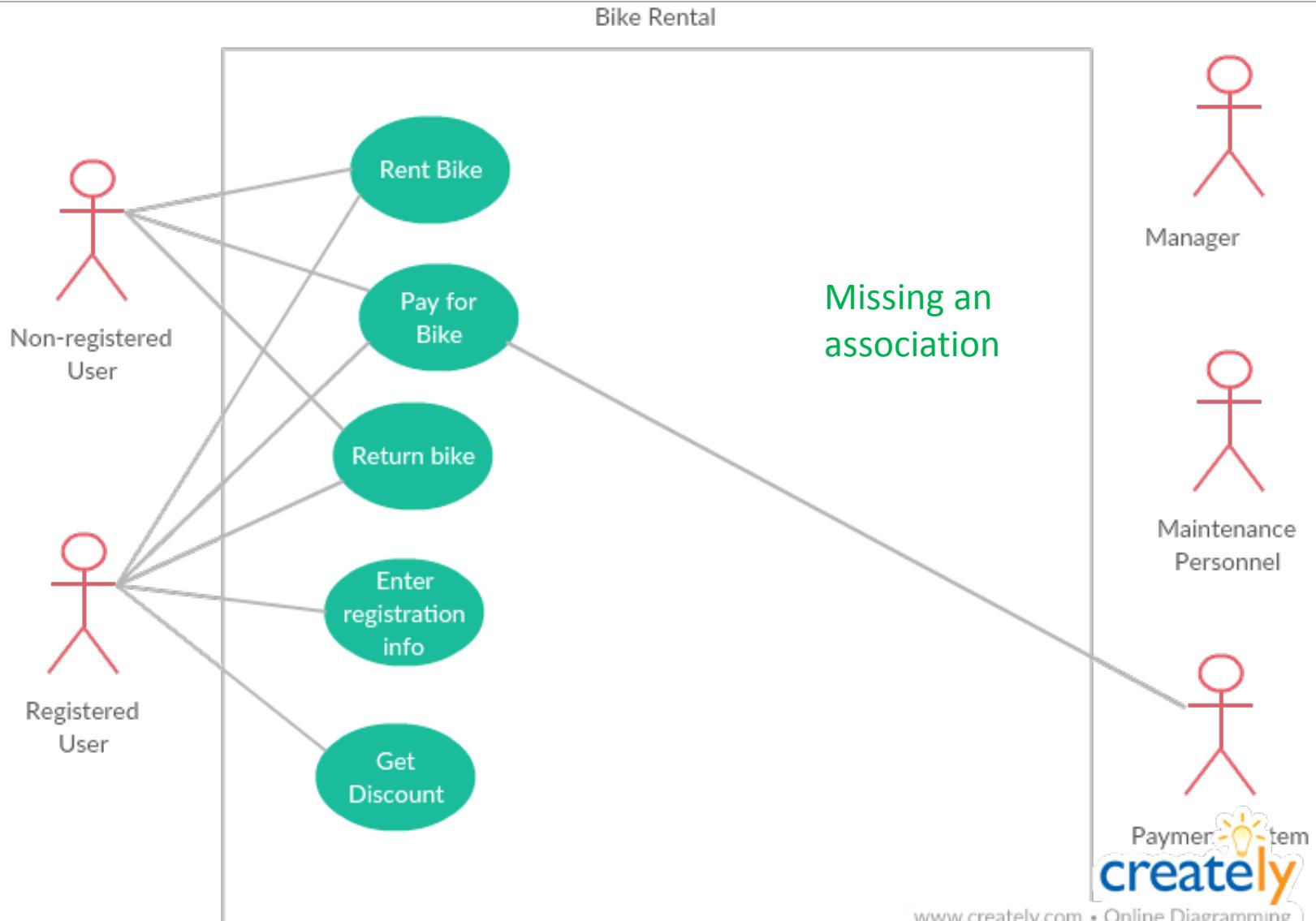
Bike Rental



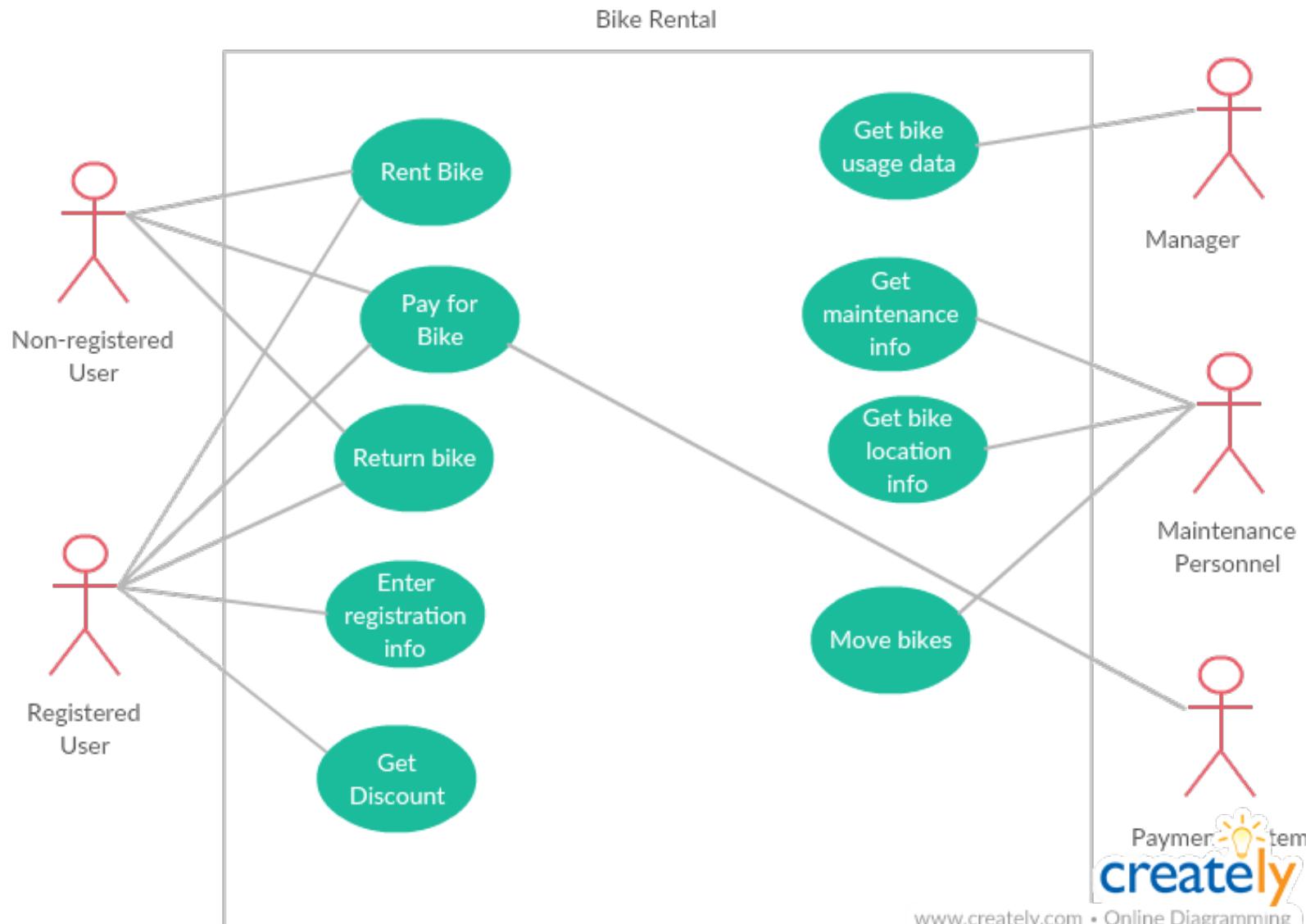
Bike Rental



Bike Rental



Bike Rental

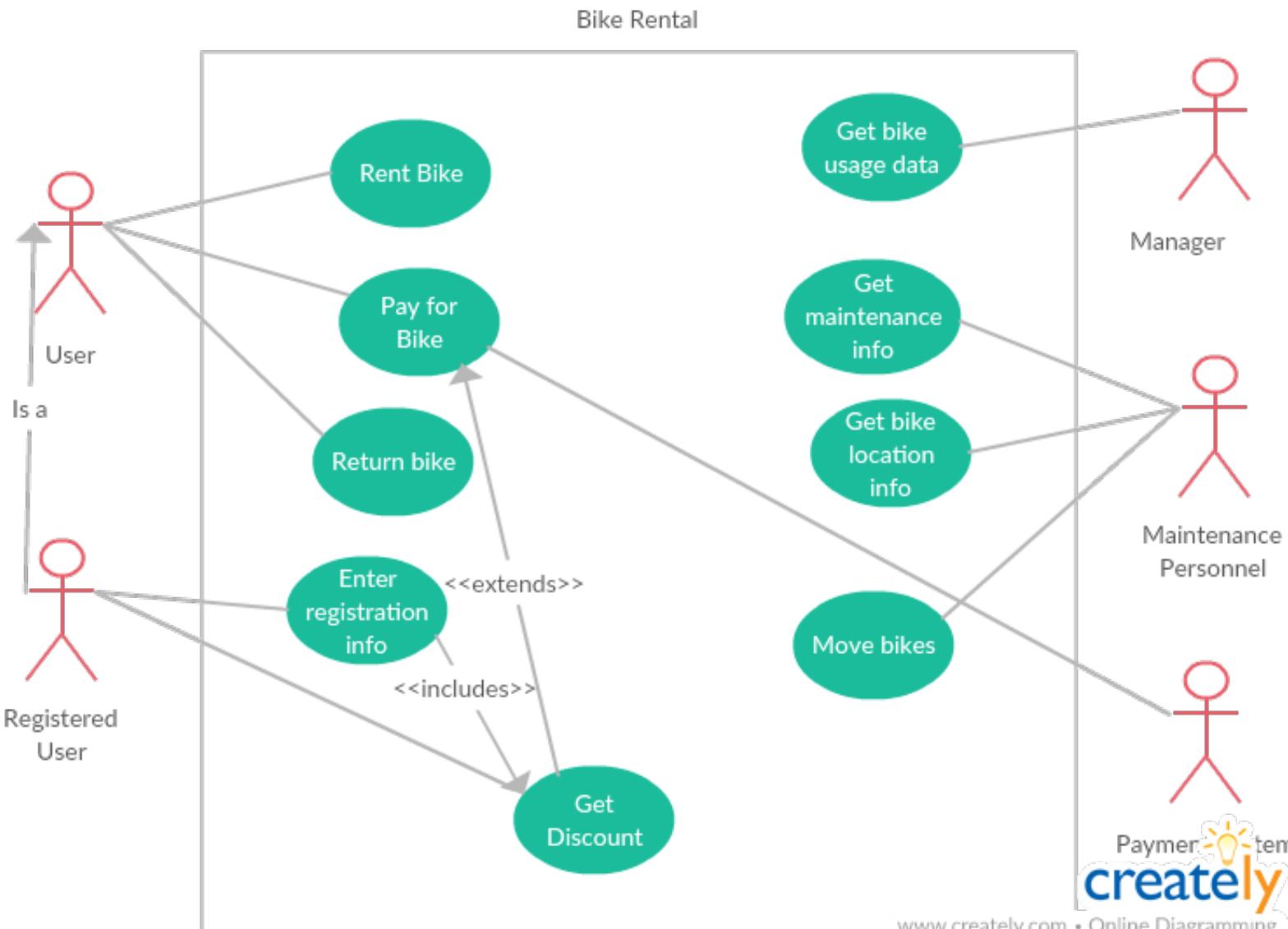


Use Cases: Advanced Concepts

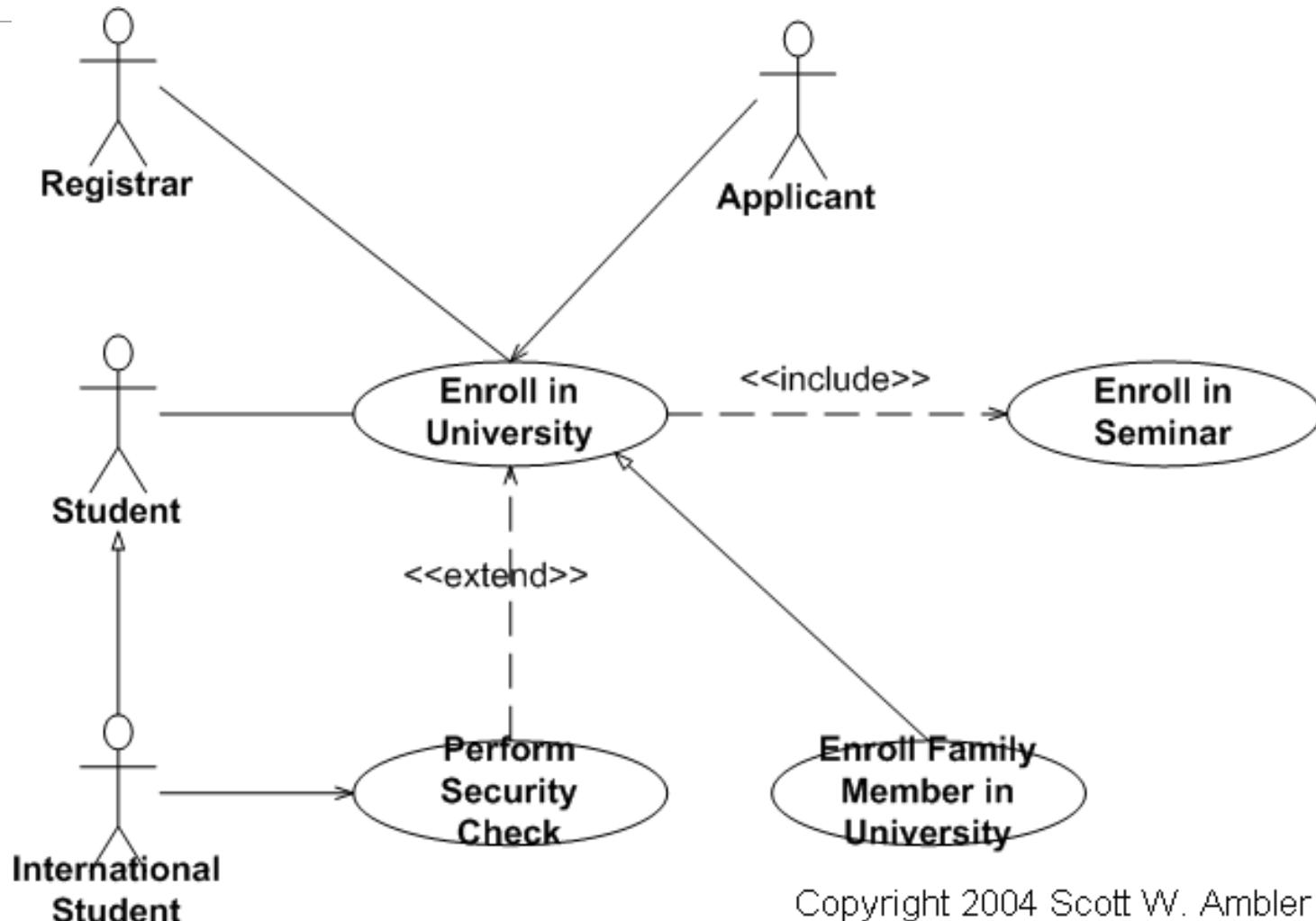
- Use cases can extend other use cases
 - “An extending use case is, effectively, an alternate course of the base use case.”
- Use cases can include other use cases
 - “the invocation of a use case by another one.”
- Actors can have an “is-a” relationship

<http://agilemodeling.com/essays/useCaseReuse.htm>
- You can use these concepts if you want, but you don’t have to.
- A1 “Identify the major use cases and relationships, both between the stakeholders and the use cases, (YES)
• and, if relevant, between use cases.” (OPTIONAL)

Bike Rental



More Examples



Copyright 2004 Scott W. Ambler

- More: <https://www.uml-diagrams.org/use-case-diagrams-examples.html>

Goal Models



Why Goal-Orientation?

- Most systems today are socio-technical
 - E.g., online commerce, healthcare, government
- Complex web of relationships among systems and stakeholders
 - Help each other achieve what they want
- Help stakeholders understand their needs:
 - E.g., security, privacy, trust, profitability, market positioning, strategic alliances, intellectual property, ...
- Understanding “why?”, not just “what?” or “how?”

Goal-Oriented Requirements Engineering

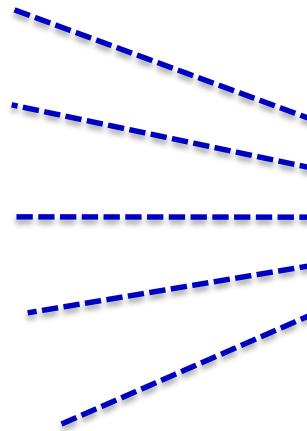
- **Captures:**
 - stakeholder needs (goals)
 - interrelationships (dependencies)
 - relationships with technology
 - alternative requirements, and
 - tradeoffs amongst alternatives
- **Benefits:**
 - Abstraction of complex problems from a social perspective
 - Communication
 - Convergent understanding of problems and solutions
 - Reasoning: trade-off analysis, “simulation”, solution-finding
 - Supports decision making
 - Captures rationale for decisions
- **Demonstrate via examples...**

Example Application: Security Requirements for a WiFi-based Navigation System

- In a WiFi-based navigation system, users navigate inside buildings using WiFi access points
- The system could use existing WiFi access points or could install new access points
- Using existing access points saves money



Location estimation



Security issues: WiFi access points may be switched off

Example Application: Strategic Requirements Analysis for Kids Help Phone

- Kids Help Phone (KHP) is a not-for-profit organization which provides counseling for Canadian children and youth.
- Traditionally, KHP has provided counseling via phone
- As new technology is introduced, KHP wanted to go where the kids are: the web
- How can counseling services be effectively provided on-line?
- How can the organization continue to ensure:
 - Anonymity? Confidentiality? Quality of Service?



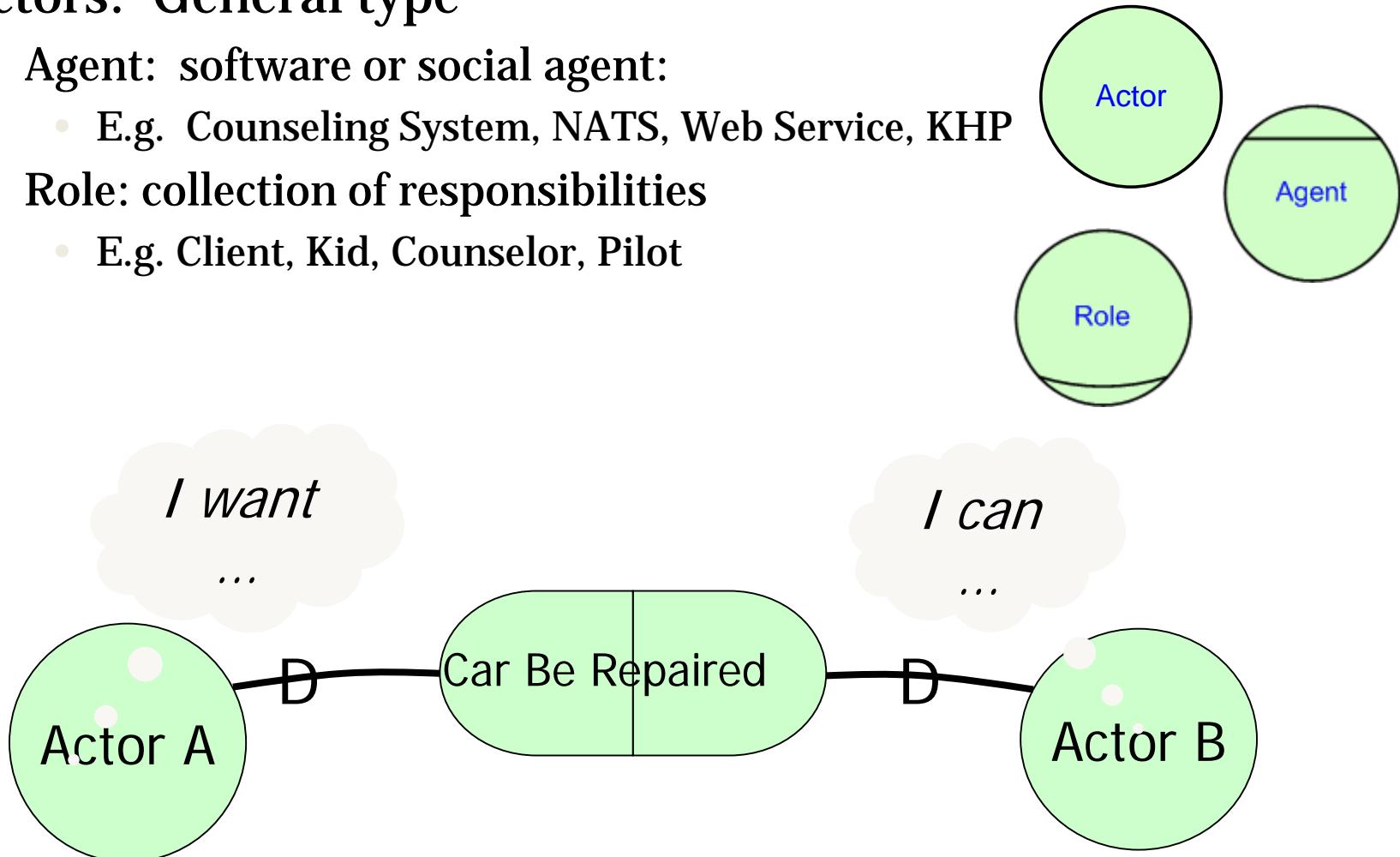
Example Application: UK Air Navigation Service Provider

- A new design is needed for the Controlled Airspace Infringement Tool (CAIT)
 - provides air traffic controllers with timely warnings of airspace infringements
- How do we know the new requirements are safe?
- How do we know how the new requirements effect the wider socio-technical system?



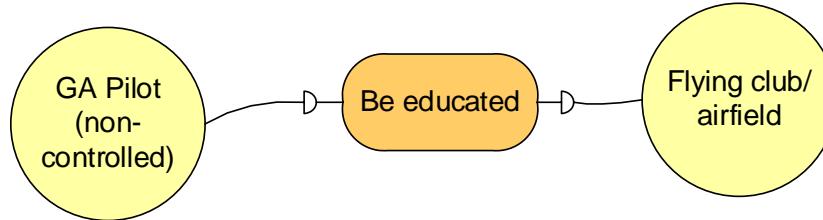
Example Goal Modeling Framework: i* (Distributed Intentionality)

- **Actors:** General type
 - **Agent:** software or social agent:
 - E.g. Counseling System, NATS, Web Service, KHP
 - **Role:** collection of responsibilities
 - E.g. Client, Kid, Counselor, Pilot

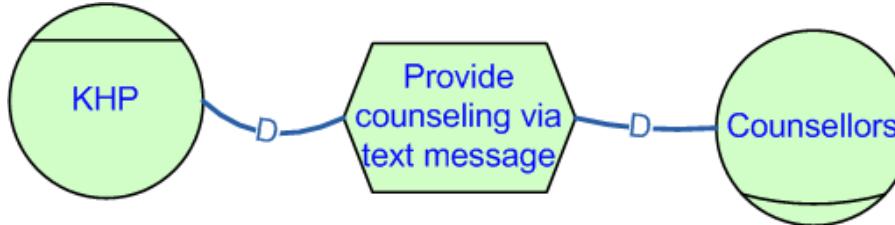


i* Strategic Dependencies

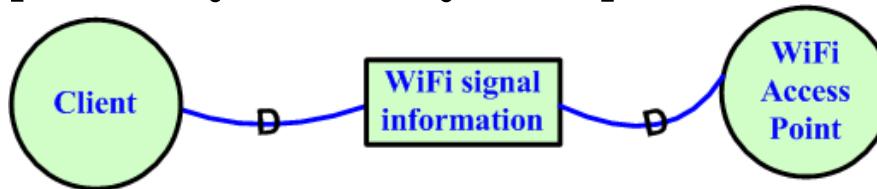
- Goal Dependency: I want you to achieve my goal, I don't care how



- Task Dependency: I want you to achieve this task, in an agreed upon way



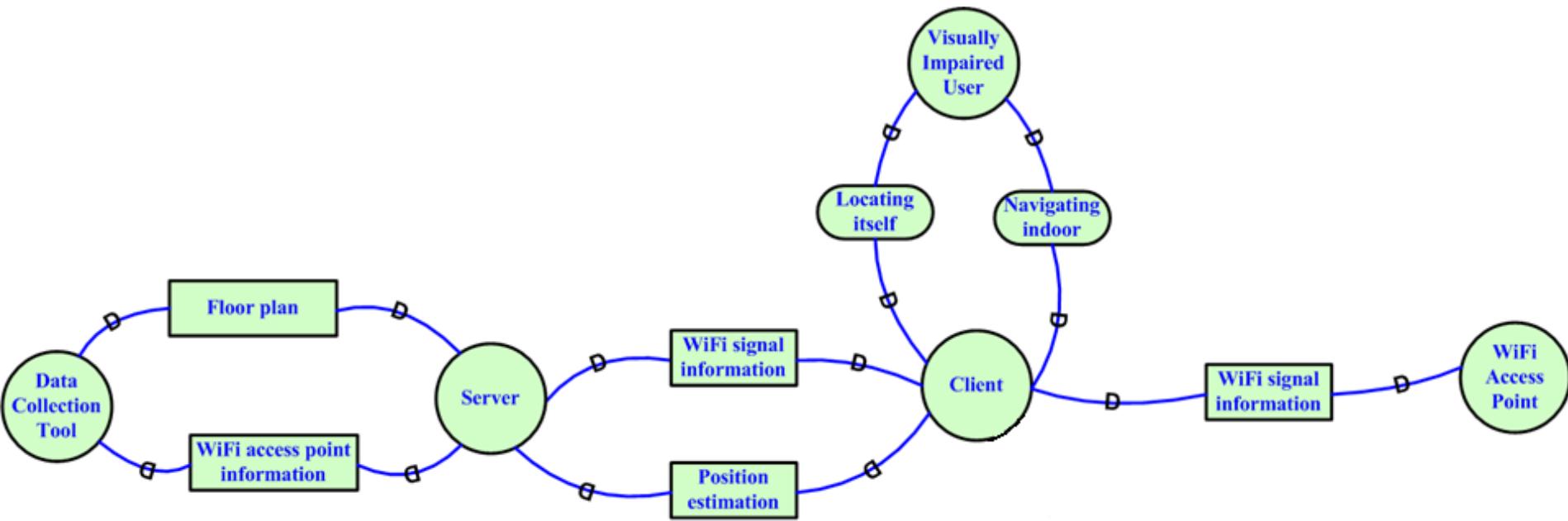
- Resource Dependency: I want you to provide this thing (entity)



- Quality Dependency: I want you to achieve my quality

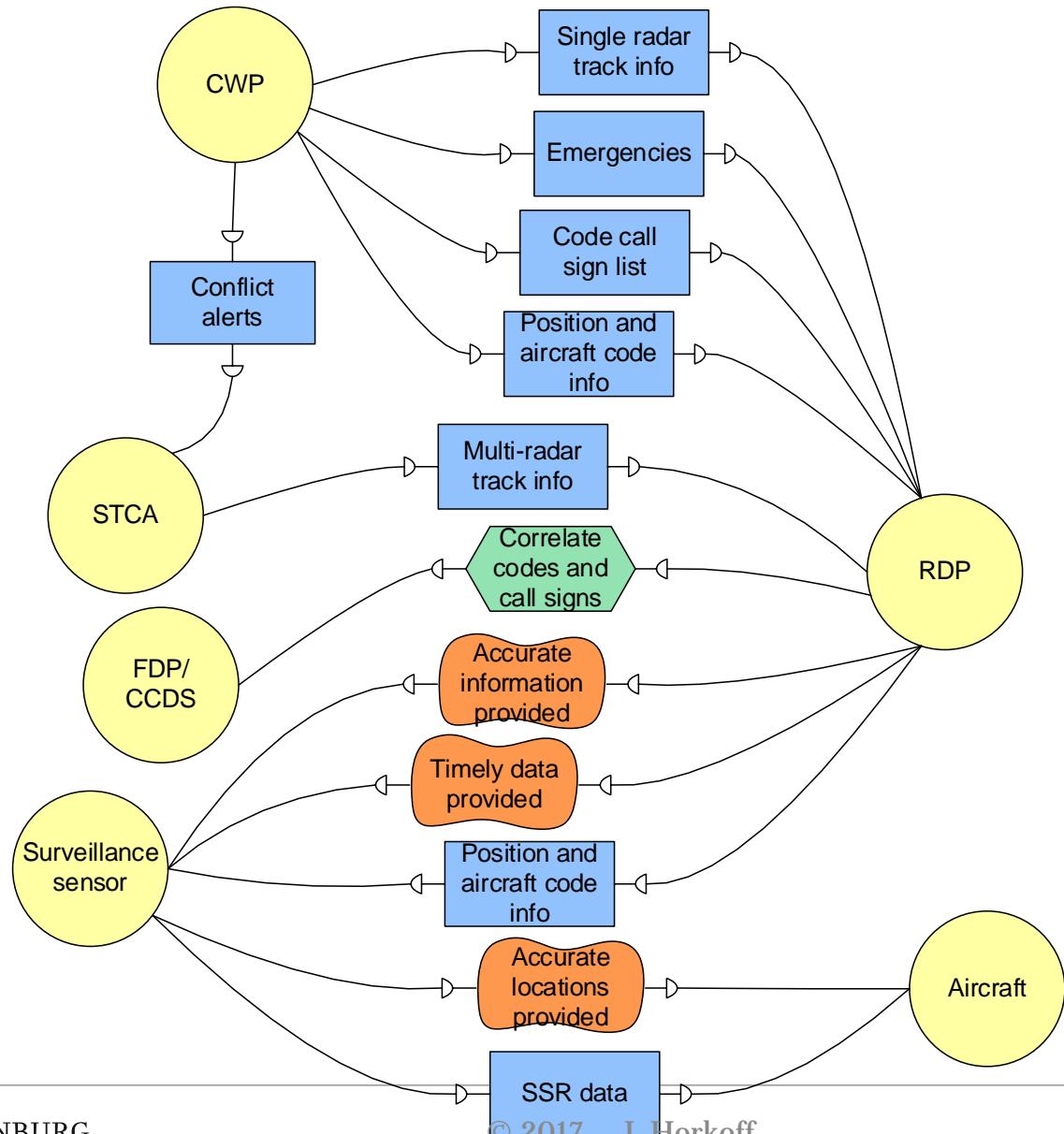


SD Example: WiFi-based Navigation System



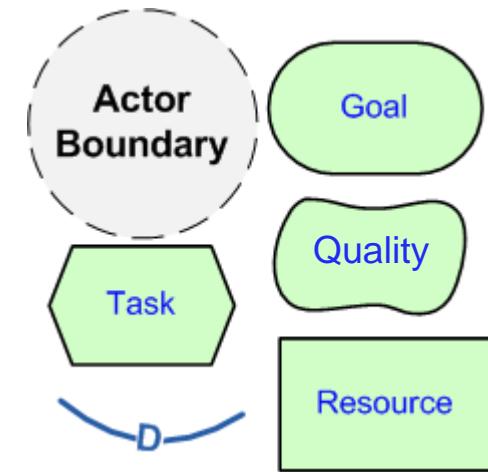
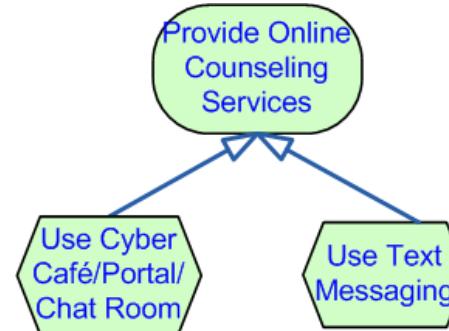
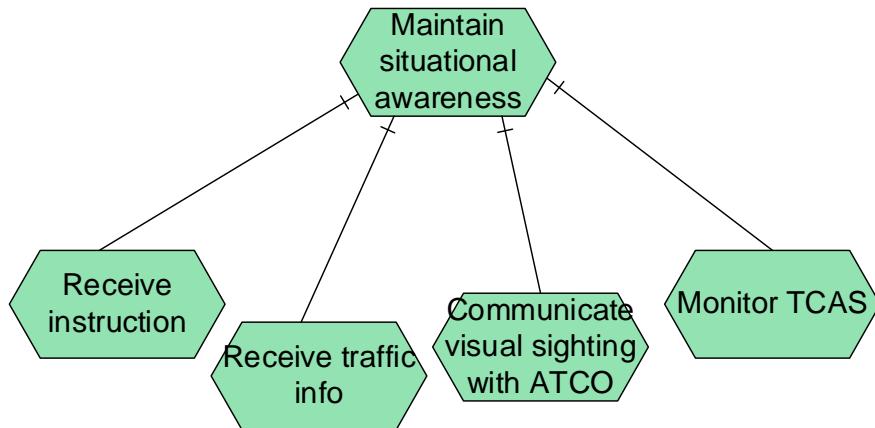
SD Example: NATS

- CWP = Controller Working Position
- STCA = Short Term Conflict Alter
- FDP/CCDS = Flight Data Processor/Code Callsign Distribution System
- RDP = Radar Data Processing

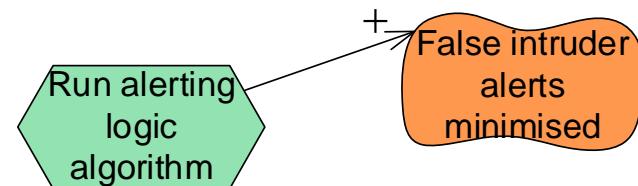


i* Strategic Rationale (SR) Diagrams

- Actor boundaries
- Goals, Qualities, Tasks, Resources
- Dependencies (as before)
- Refinement (AND), (OR)

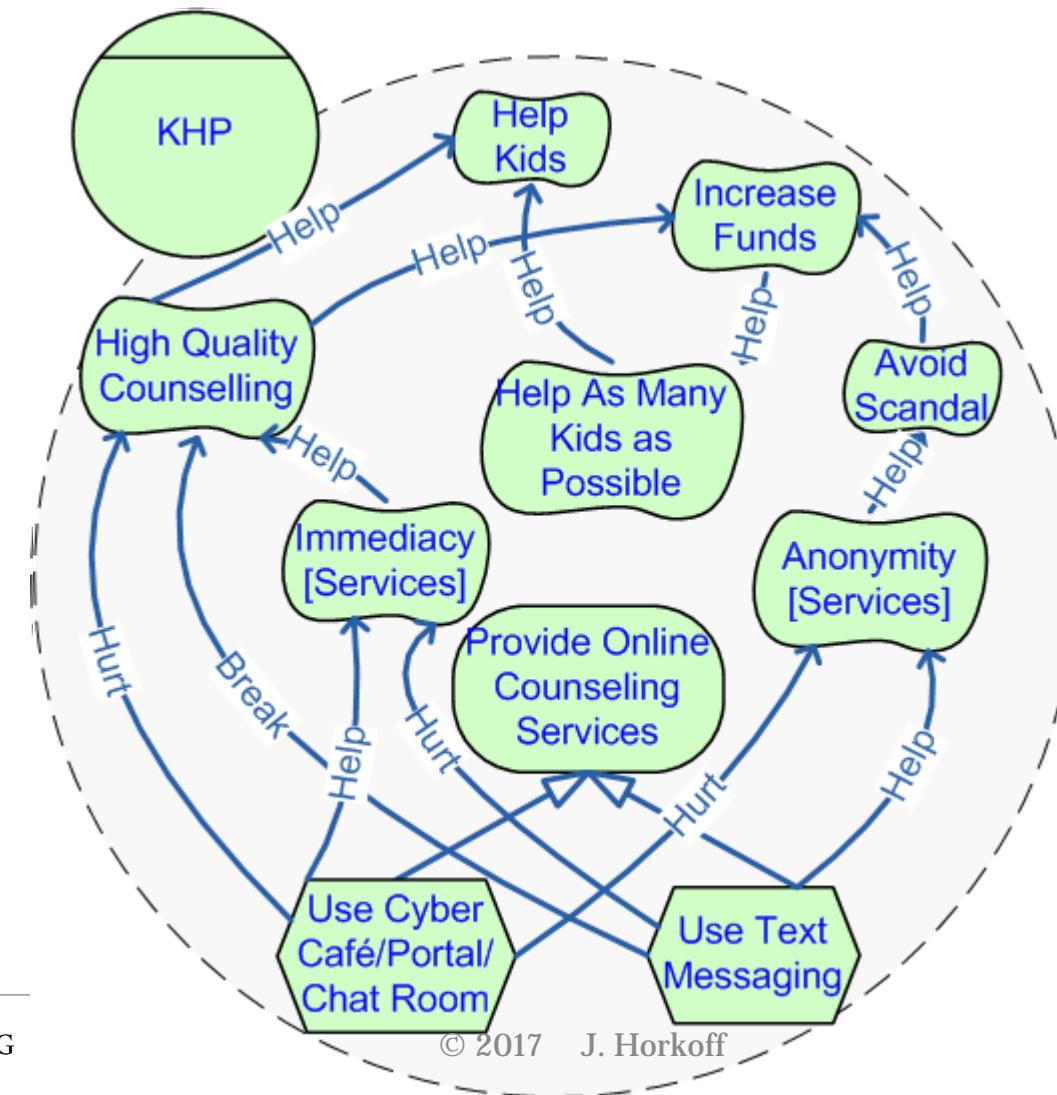


- Contribution: Make, Help (+), Hurt (-), Break

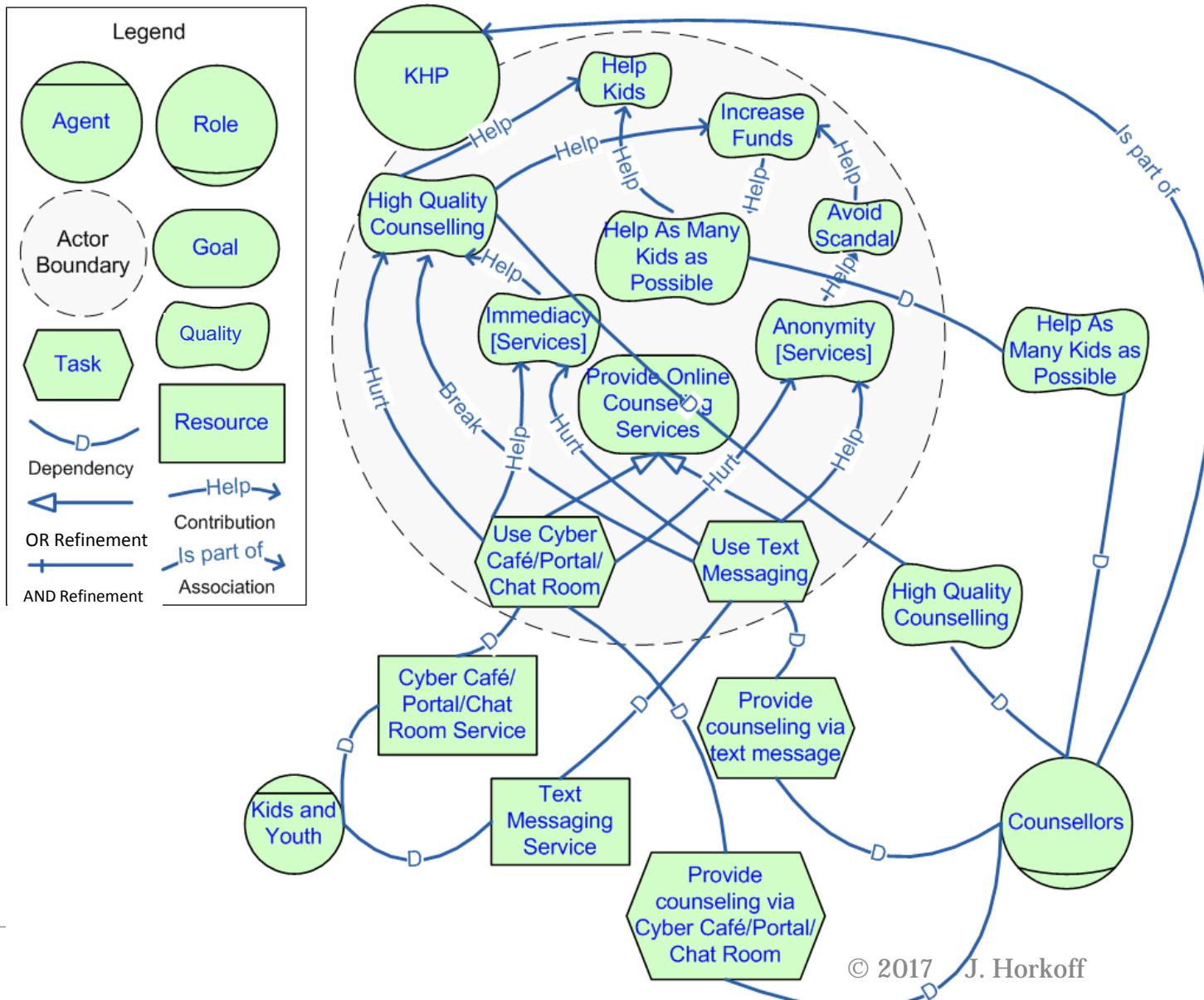


SR Example: Kids Help Phone

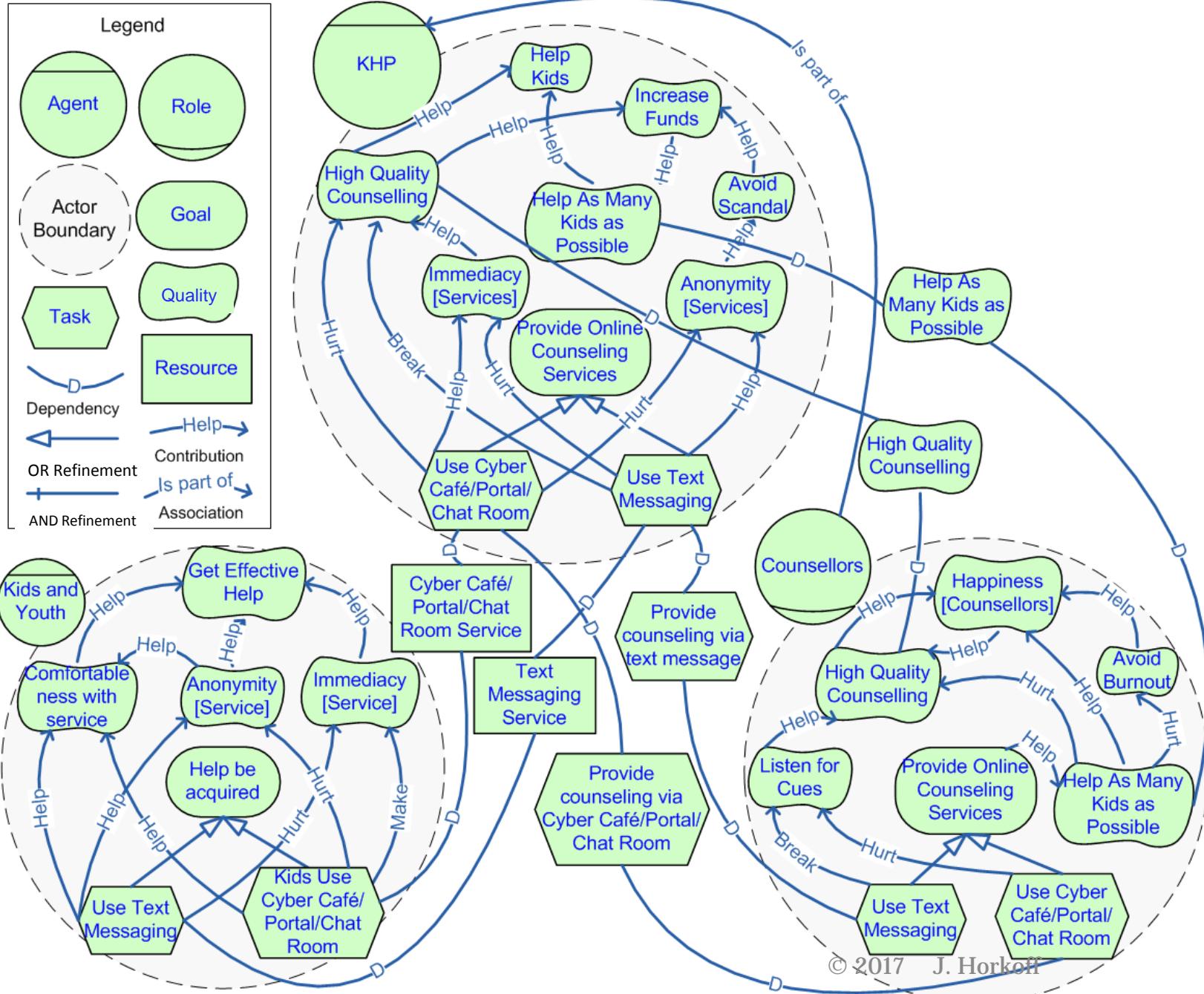
- What are the goals of KHP?
- What are the relationships between the goals?
- What are the alternatives?
- How can these goals be satisfied?
- What are the negative consequences?
- Who do we depend on?
- For what?
- How?



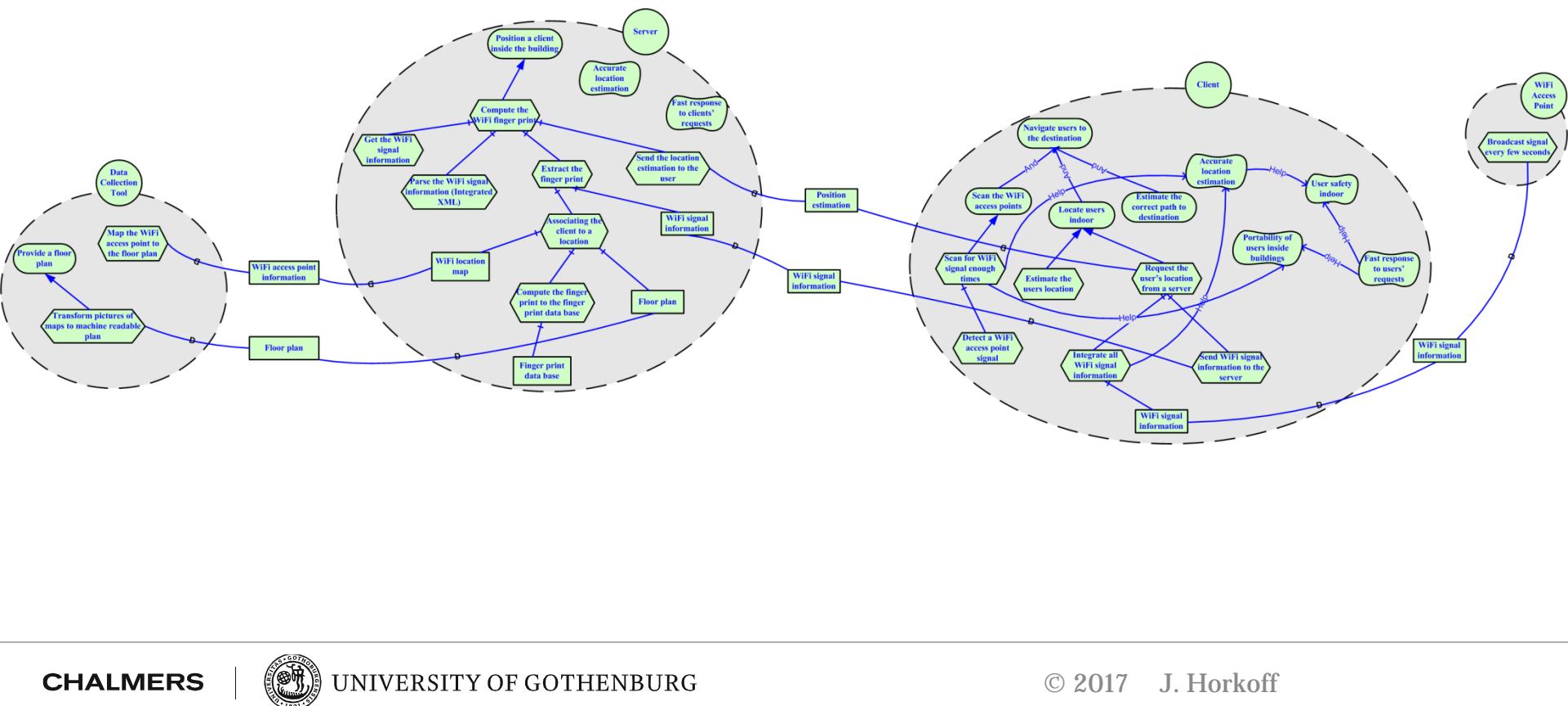
SR Example: Kids Help Phone



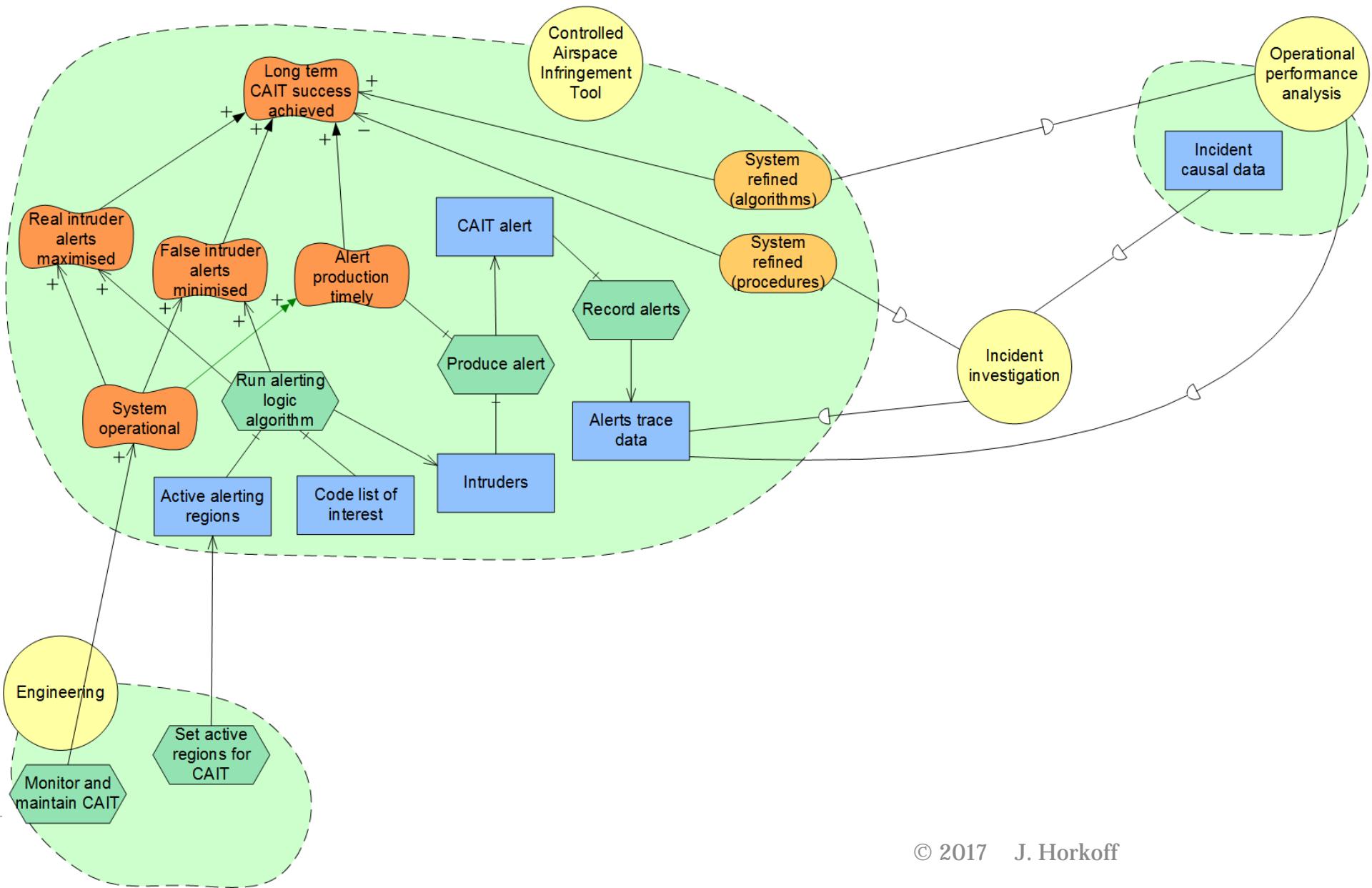
CSR Example: Kids Help Phone



SR Example: Wifi (high-level overview)



SR Example: NATS (partial)



iStar (i^*)

- There's more than one goal modeling language
- We'll use iStar 2.0
- Evolution of i^* (examples you've seen so far)
- More info can be found (Document also on GUL)
 - <https://arxiv.org/abs/1605.07767>
 - Additional slides on GUL for university travel example

Creative Leaf

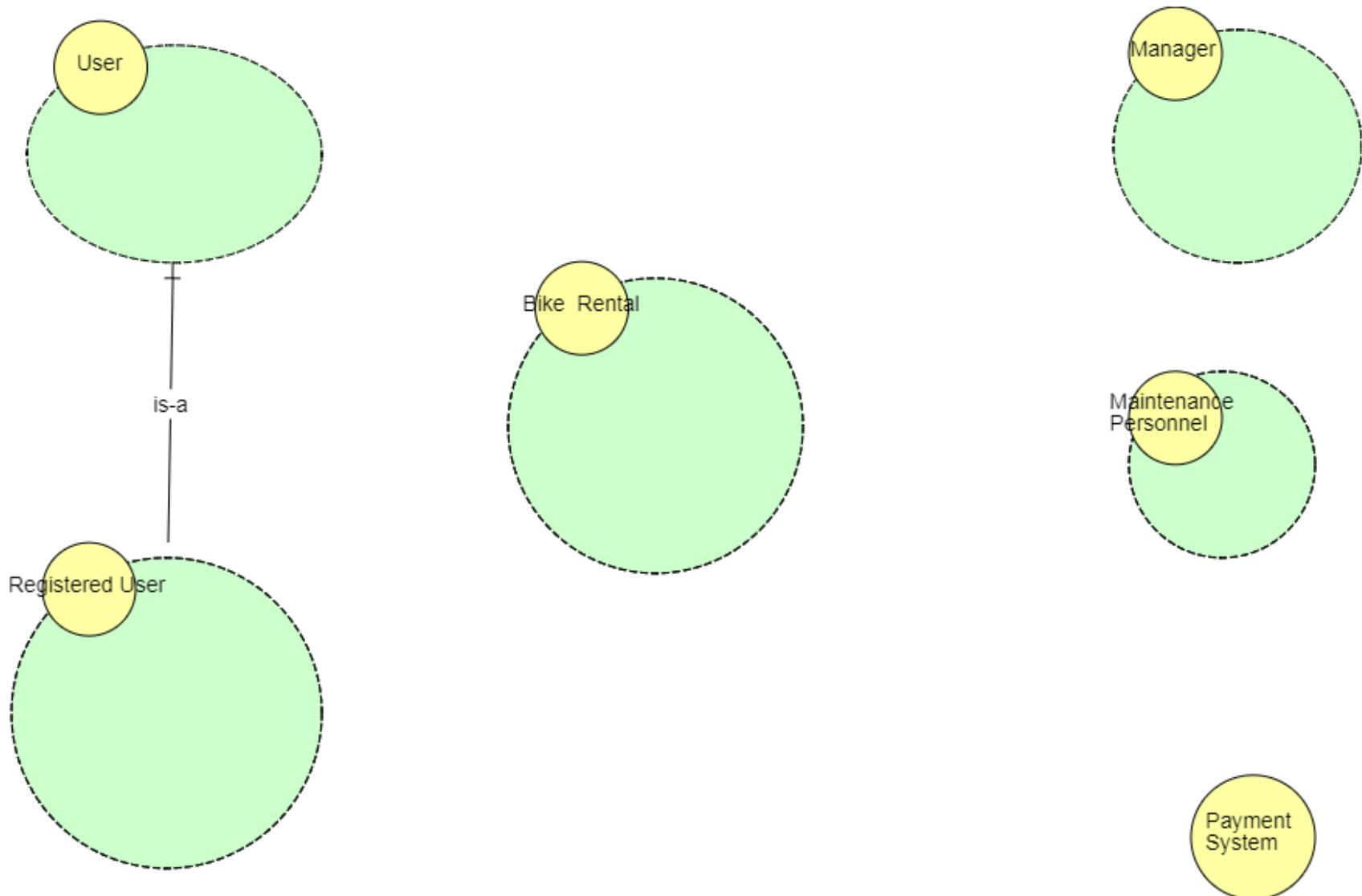
- Web-based goal modeling tool
- Use in Chrome (!)
- <http://creativeleaf.portal.chalmers.se/>
- (backup site: <http://creativeleaf.city.ac.uk/>)
- Former has syntax restrictions, latter does not, i.e., you can do more wrong things in the UK version, but have more freedom.
- Bugs? Email me: jenho@chalmers.se

Creative Leaf

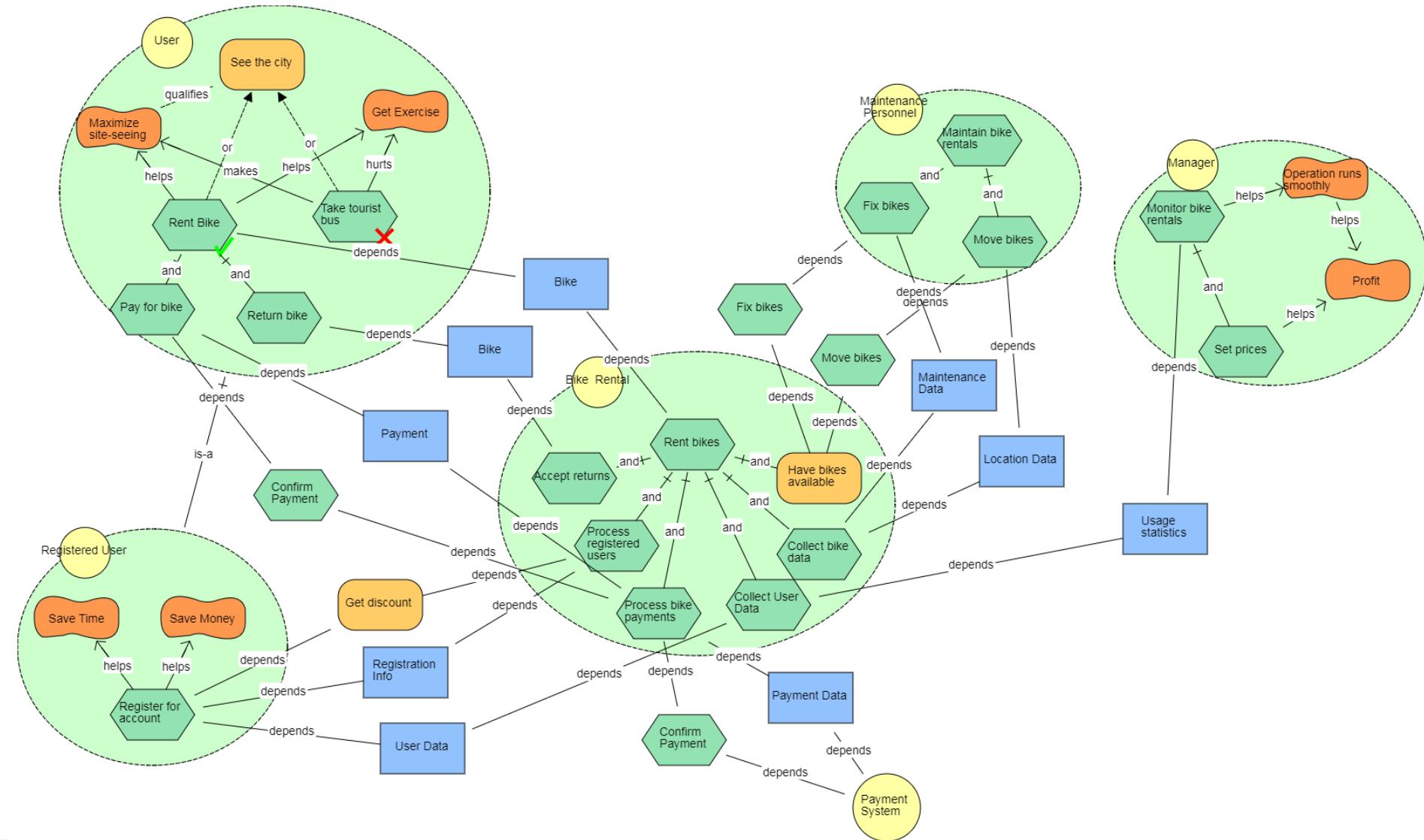
- <http://creativeleaf.portal.chalmers.se/>

The screenshot shows the Creativeleaf portal interface. At the top, there's a navigation bar with links for Open as: SVG, PNG, Clear, Save, Load, Font Size (+ -), About, Options, iStar Help, Legend, Creativity Help, Ideas, and Hover Evaluate. The main area features a 'Stencil' library on the left containing icons for Goal (orange rounded rectangle), Task (green hexagon), Quality (orange rounded rectangle), Resource (blue rectangle), Actor (yellow circle), and another Actor icon partially visible inside a green dashed oval. To the right is a workspace with a dotted grid background. On the far right, there's a sidebar with sections for BRAINSTORM (IDEA icon), CRUISE (text '» CRUISE'), PAIRWISE COMPARISON (scales icon), CREATIVITY TRIGGERS (bulb icon), and Bright Sparks (lightning bolt icon). At the bottom left, there's a footer with City University London branding, copyright information (2017), and credits for Leaf, client:IO, JointJS, and FlatIcon.com.

Bike Rental: Actors



Bike Rental: All

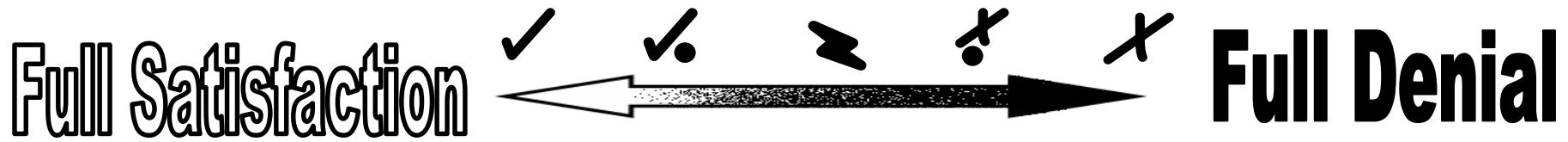


Reasoning with Goal Models

- Analysis procedures can answer questions, e.g.,
 - What if a particular alternative is selected? How does this effect goals and actors in a model?
 - If I want to satisfy a (set of) goal(s), what alternatives should I select? What tasks should I implement?

Reasoning Example Procedure

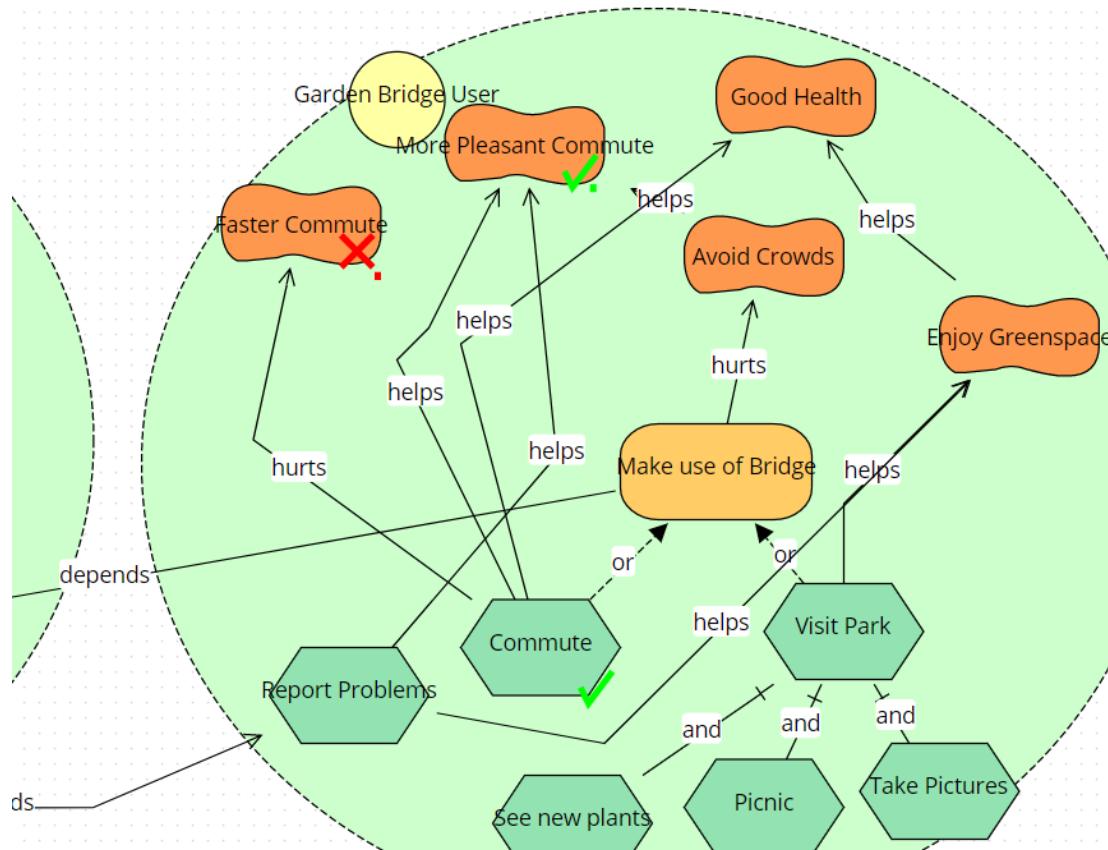
- We pick an example approach for illustration:
qualitative, interactive analysis, [Horkoff & Yu, REJ'14]
- Use qualitative labels to represent degree of satisfaction



- Labels throughout the model using propagation rules

Creative Leaf Hover Evaluate

Save Load Font Size + - About Options iStar Help Legend Creativity Help Ideas Hover Evaluate



Approximate Model Mappings

Context Diagram	Use Cases	Goal Models
System actor	System boundary	System actor
Other actors	Actors	Actors
Inputs/Outputs	Roughly map to use cases	Dependencies
	Use Case	(Usually) Task
		Qualities
		And/Or Refinement
		Contribution

Questions?

