

Distributed Systems Modelling

Rui Carlos Oliveira

Universidade do Minho

(rco@di.uminho.pt)

Based on Fred B. Schneider's "What Good are Models and What Models are Good?"





- Experimental observation
- Build things and observe how they behave in various settings.
- Experience enables us to build things for settings similar to those that have been studied.



- Experimental observation
 - Build things and observe how they behave in various settings.
 - Experience enables us to build things for settings similar to those that have been studied.
- Modelling and analysis
 - Formulate a model by simplifying the object of study and postulating a set of rules to define its behaviour.
 - Analyze the model and infer consequences.



- Experimental observation
- Build things and observe how they behave in various settings.
- Experience enables us to build things for settings similar to those that have been studied.
- Modelling and analysis
 - Formulate a model by simplifying the object of study and postulating a set of rules to define its behaviour.
 - Analyze the model and infer consequences.

Practice

- Experimental observation
 - Build things and observe how they behave in various settings.
 - Experience enables us to build things for settings similar to those that have been studied.
- Modelling and analysis
 - Formulate a model by simplifying the system of study and postulating a set of rules for its behaviour.
 - Analyze the model and infer properties.



Practice



Theory





- Modeling a system consists of identifying and precisely characterizing the system's entities and the behavioral rules governing them that are relevant for the reasoning or problems at hand



- Modeling a system consists of identifying and precisely characterizing the system's entities and the behavioral rules governing them that are relevant for the reasoning or problems at hand
- A model needs to be tractable in that it shouldn't encompass uninteresting details that would impair the desired analysis and, at the same time, it needs to be accurate so that it captures all attributes affecting the phenomena of interest.





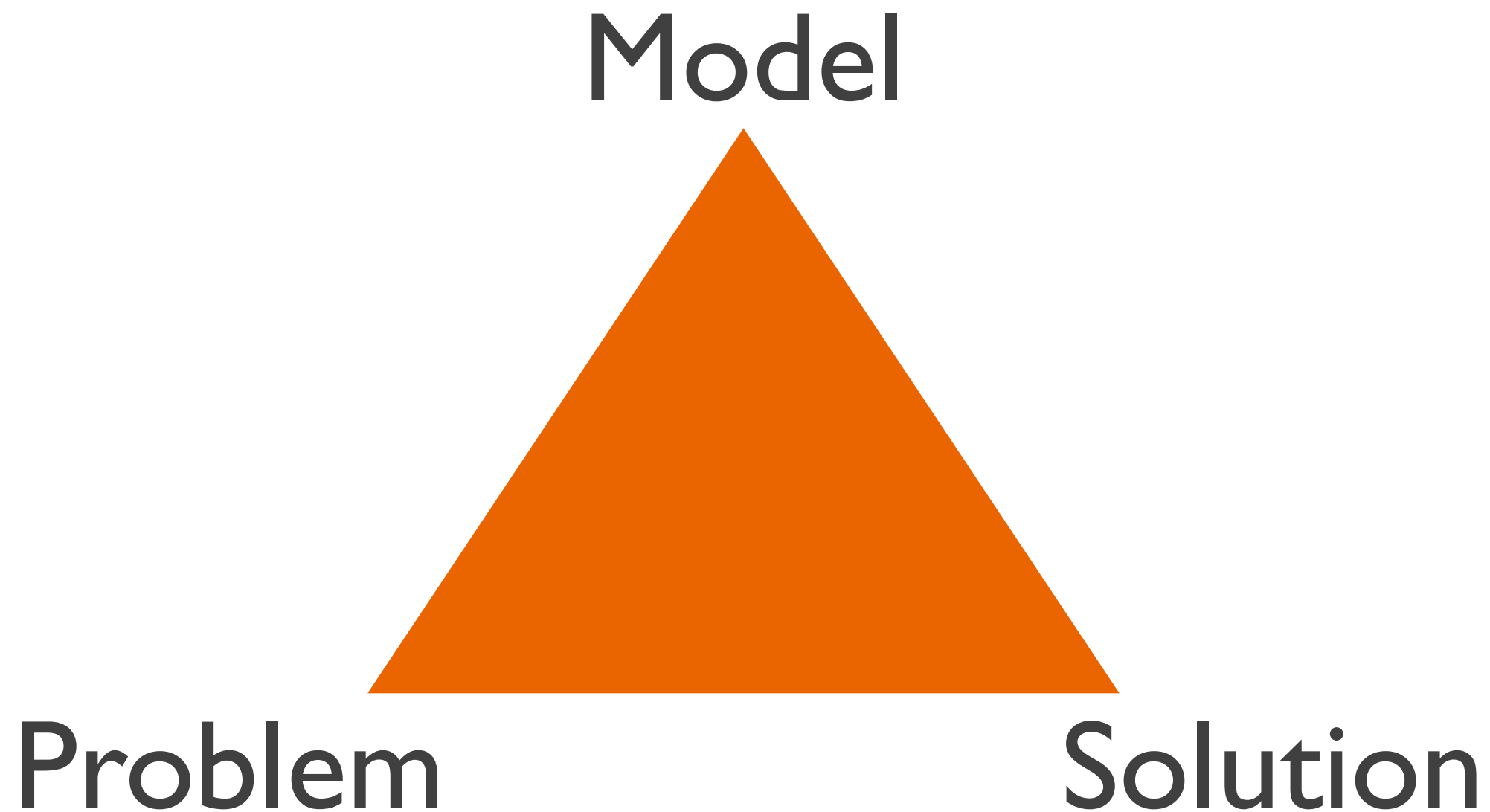


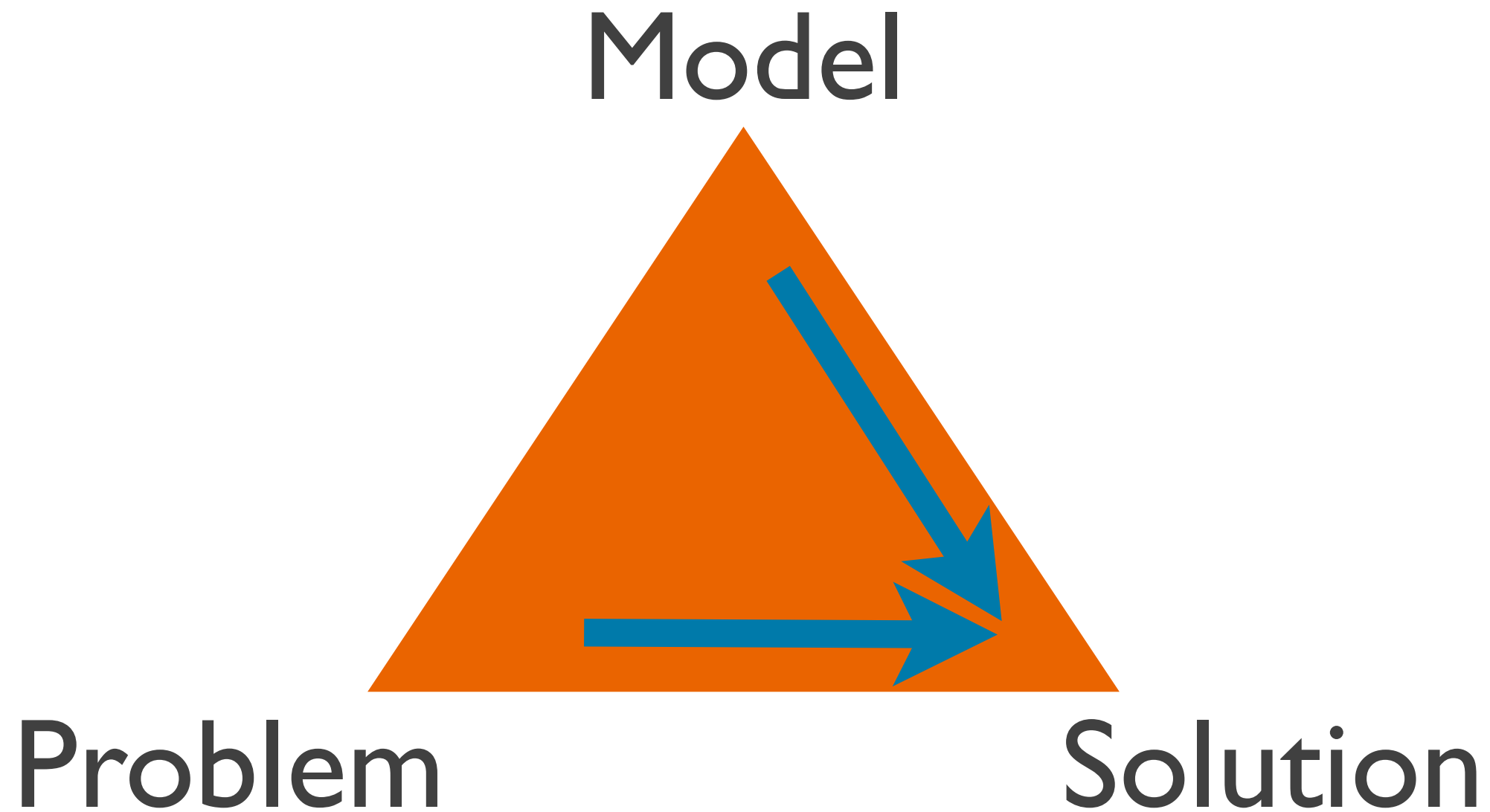
- With a model for a distributed system model we seek answers to two fundamental questions:
- Feasibility: what classes of problems can be solved?
- Cost: how expensive must the solution be? What's the relative cost of the solutions?



- With a **model** for a distributed system model we seek answers to two fundamental questions:
- Feasibility: what classes of **problems** can be solved?
- Cost: how expensive must the **solution** be? What's the relative cost of the solutions?









- A minimal model for a distributed system usually consists of a finite set of autonomous processing entities connected through communication channels.



- A minimal model for a distributed system usually consists of a finite set of autonomous processing entities connected through communication channels.
- Typically, we model a distributed system as:
 - a finite set of processes
 - a finite set of communication channels
 - an adversary
 - a set of rules that govern the behavior of the attributes and the power of the adversary



Feasibility: an example



- System model:
- Two processes, A and B, communicate by sending and receiving messages
- Neither process can fail. However, the channel can experience transient failures, resulting in the loss of a subset of the messages that have been sent



- System model:
 - Two processes, A and B, communicate by sending and receiving messages
 - Neither process can fail. However, the channel can experience transient failures, resulting in the loss of a subset of the messages that have been sent
- A Coordination problem:
 - Devise a protocol where either of two actions α and β are possible, but (i) both processes take the same action and (ii) neither takes both actions.





- Of major importance when modeling a distributed system are the assumptions we make regarding the synchrony or asynchrony of its attributes



- Of major importance when modeling a distributed system are the assumptions we make regarding the synchrony or asynchrony of its attributes
- Modeling an attribute as asynchronous means that we will make no assumptions with respect to the time it takes to perform an action. This applies to both processes and communication channels



- Of major importance when modeling a distributed system are the assumptions we make regarding the synchrony or asynchrony of its attributes
- Modeling an attribute as asynchronous means that we will make no assumptions with respect to the time it takes to perform an action. This applies to both processes and communication channels
- A synchronous system model establishes bounds on the processes relative speeds and on the communication channels delays



- Of major importance when modeling a distributed system are the assumptions we make regarding the synchrony or asynchrony of its attributes
- Modeling an attribute as asynchronous means that we will make no assumptions with respect to the time it takes to perform an action. This applies to both processes and communication channels
- A synchronous system model establishes bounds on the processes relative speeds and on the communication channels delays
- An asynchronous model leads to universal solutions with respect to time.



Cost: an example



- System model:
 - A set of processes P_1, P_2, \dots, P_n . Each process P_i has a unique identifier $uid(i)$
 - There is no adversary
 - All processes start executing at the same time



- System model:
 - A set of processes P_1, P_2, \dots, P_n . Each process P_i has a unique identifier $uid(i)$
 - There is no adversary
 - All processes start executing at the same time
- An Election problem:
 - Devise a protocol so that a unique leader is selected and all of the processes learn its identity





- The characterization of the adversary can be done through the identification of the type, number and frequency of the **deviations** to the specified behavior of the attributes.

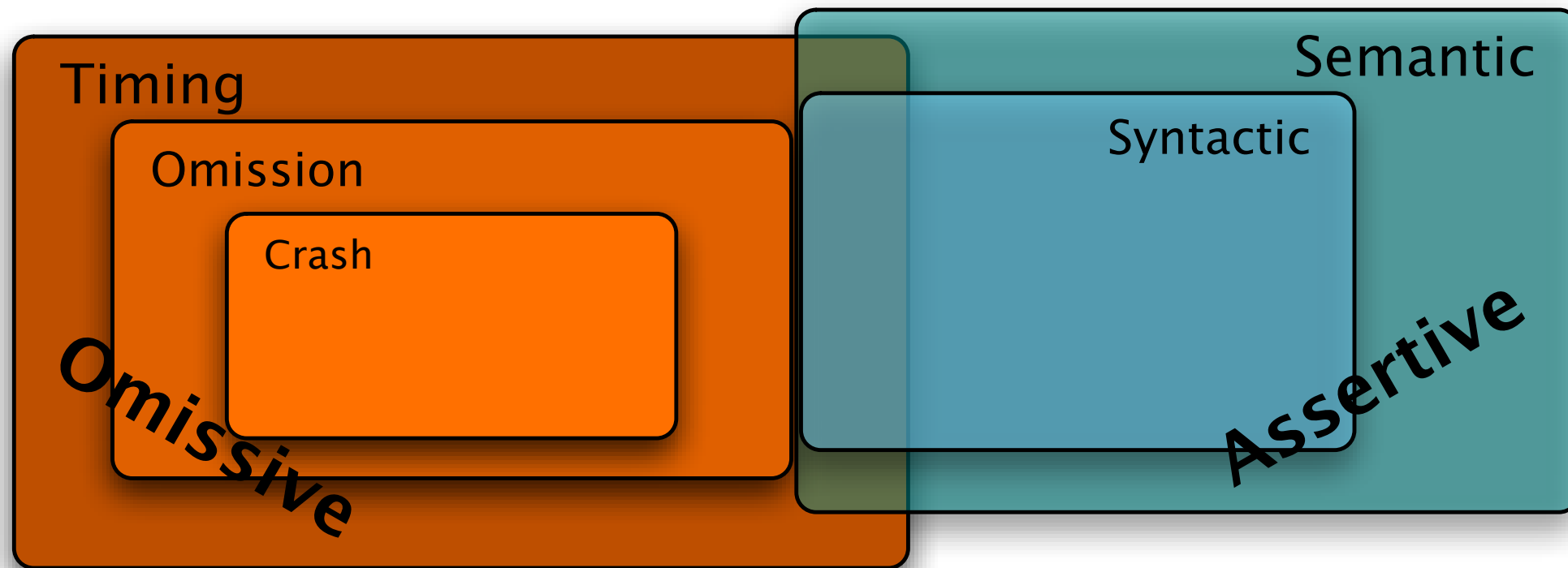


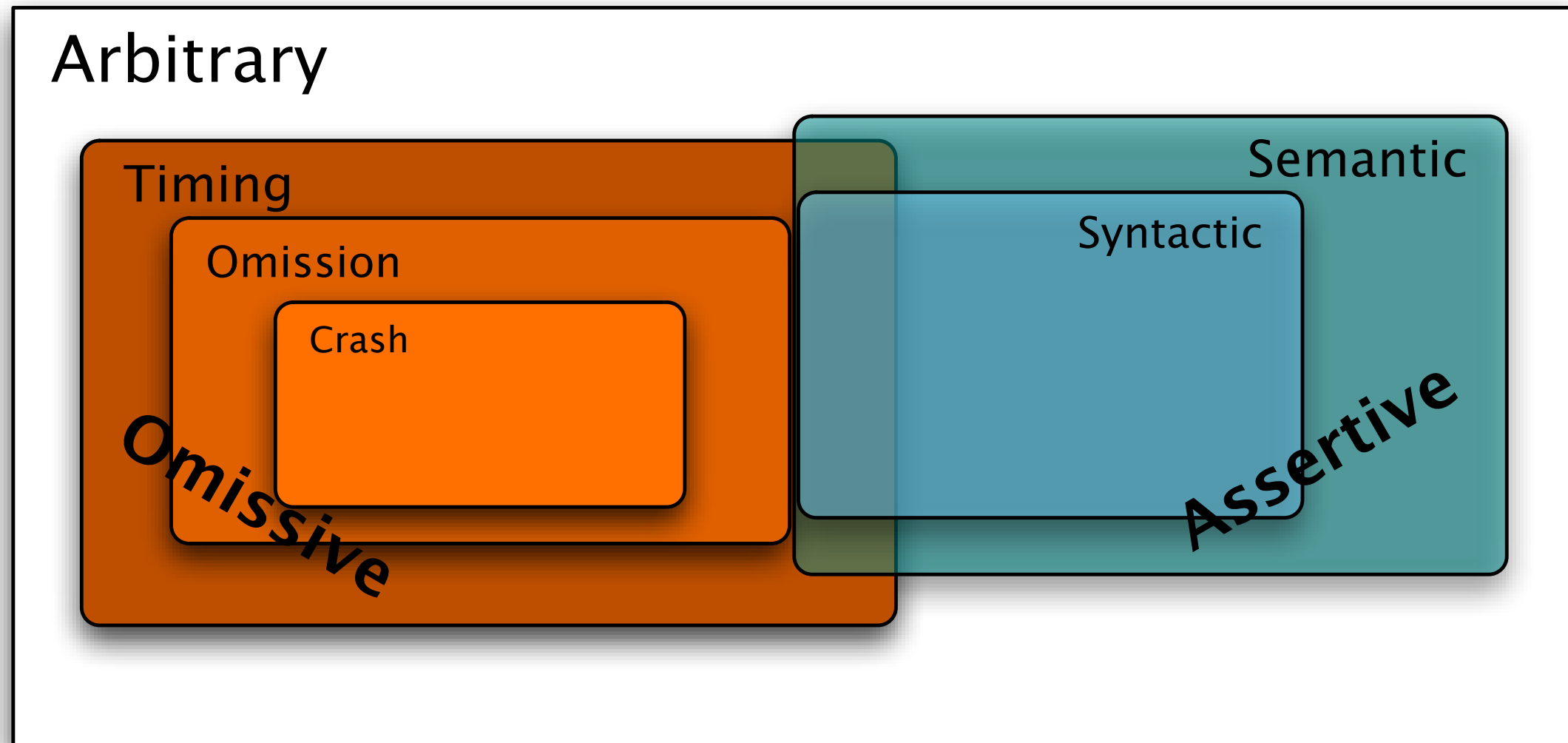
- The characterization of the adversary can be done through the identification of the type, number and frequency of the **deviations** to the specified behavior of the attributes.
- The following fault models for distributed systems can be found in literature:
 - Omissive: fail-stop, crash-stop, crash-recovery, crash-link, receive, send and general omissions
 - Assertive: Syntactic/Semantic, Byzantine



Fault classes and Coverage

Arbitrary





- Coverage: given a fault in the system, coverage is the probability that it will be tolerated



● Faults and Fault Detection



- Faults and Fault Detection
- Fault tolerance
 - Permanent vs transient faults
 - “Allowed” number of faults
 - Message loss, Process failure, Liars



- Faults and Fault Detection
- Fault tolerance
 - Permanent vs transient faults
 - “Allowed” number of faults
 - Message loss, Process failure, Liars
- Theory vs Practice
 - Worst-case scenarios, impossibilities
 - Real-world timeliness, patience and loss of temper



- Faults and Fault Detection
- Fault tolerance
 - Permanent vs transient faults
 - “Allowed” number of faults
 - Message loss, Process failure, Liars
- Theory vs Practice
 - Worst-case scenarios, impossibilities
 - Real-world timeliness, patience and loss of temper
- A model for the Internet

