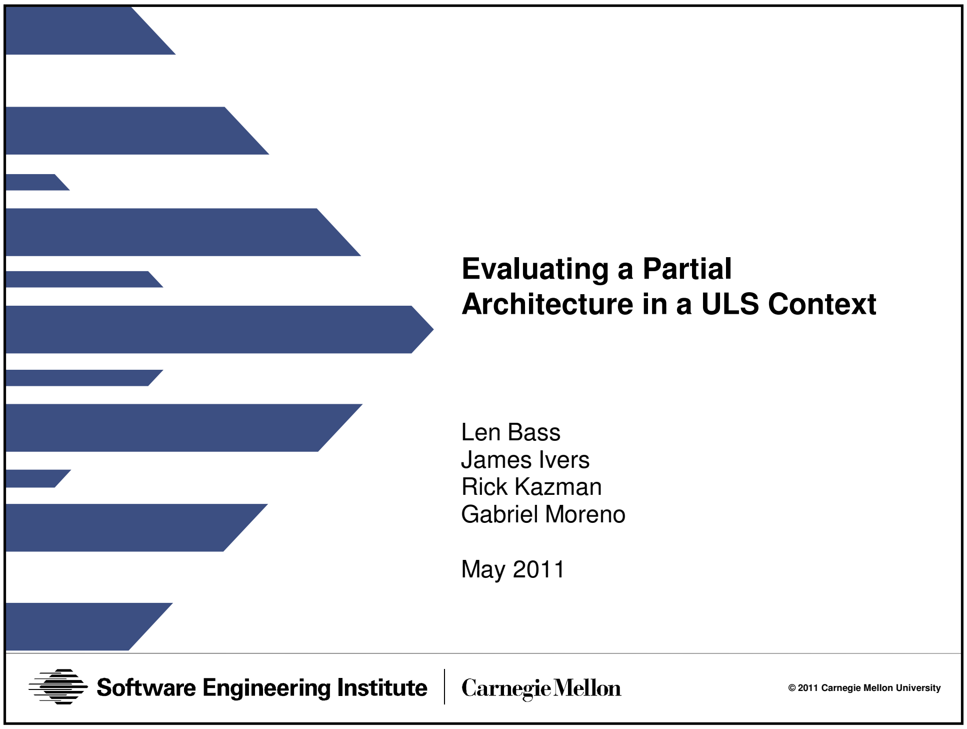
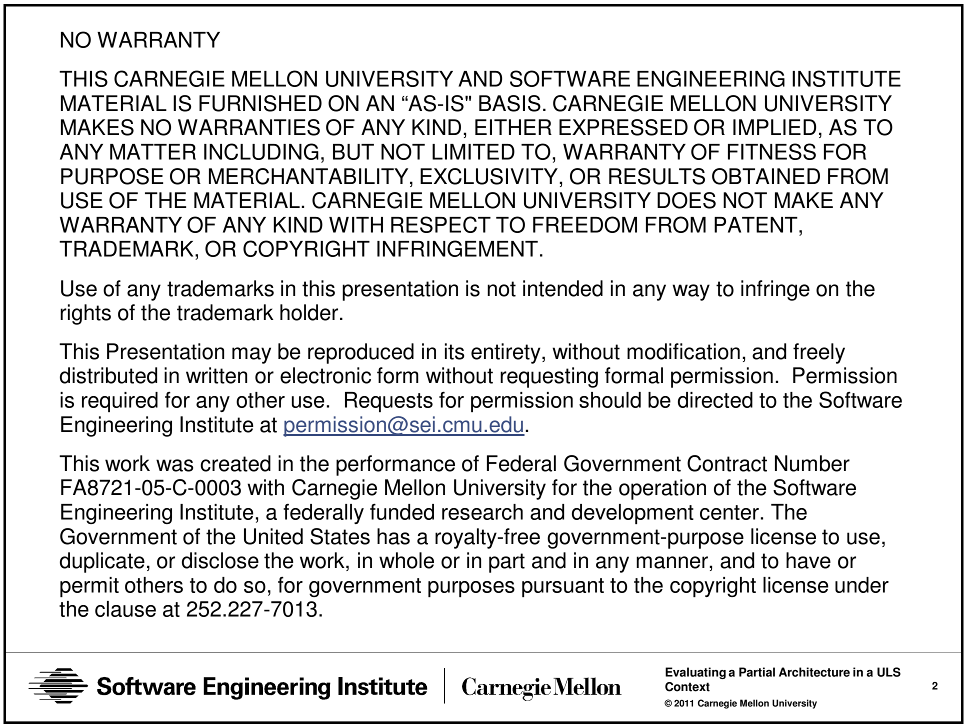


Evaluating a Partial Architecture in a ULS Context

Len Bass
James Ivers
Rick Kazman
Gabriel Moreno

May 2011

Software Engineering Institute | Carnegie Mellon



NO WARRANTY

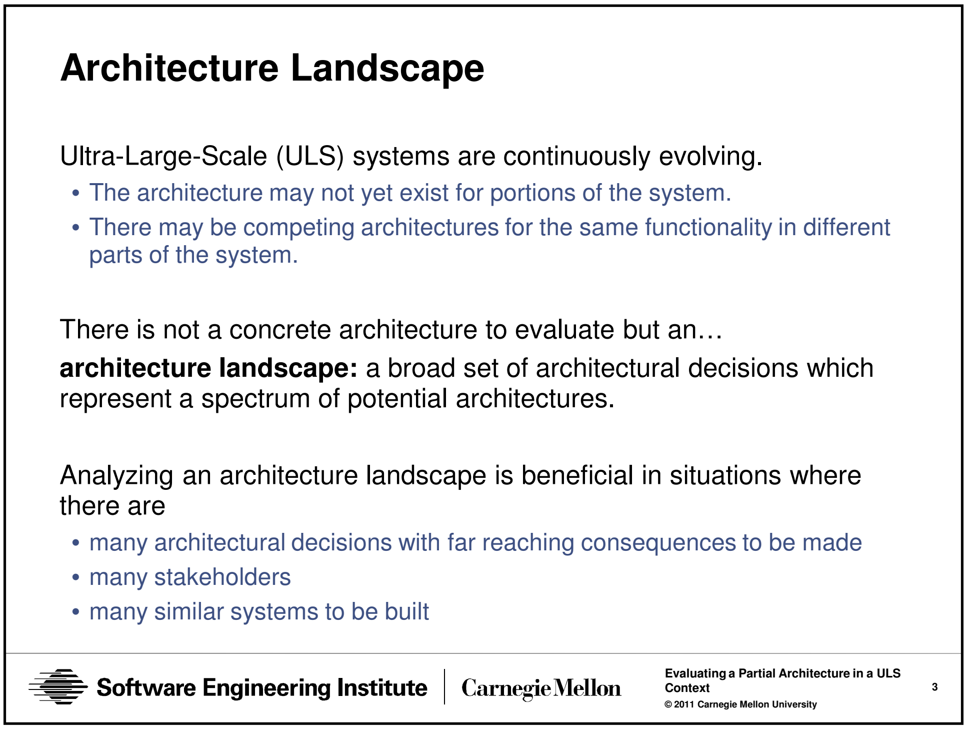
THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

Software Engineering Institute | Carnegie Mellon

Evaluating a Partial Architecture in a ULS Context 2



Architecture Landscape

Ultra-Large-Scale (ULS) systems are continuously evolving.

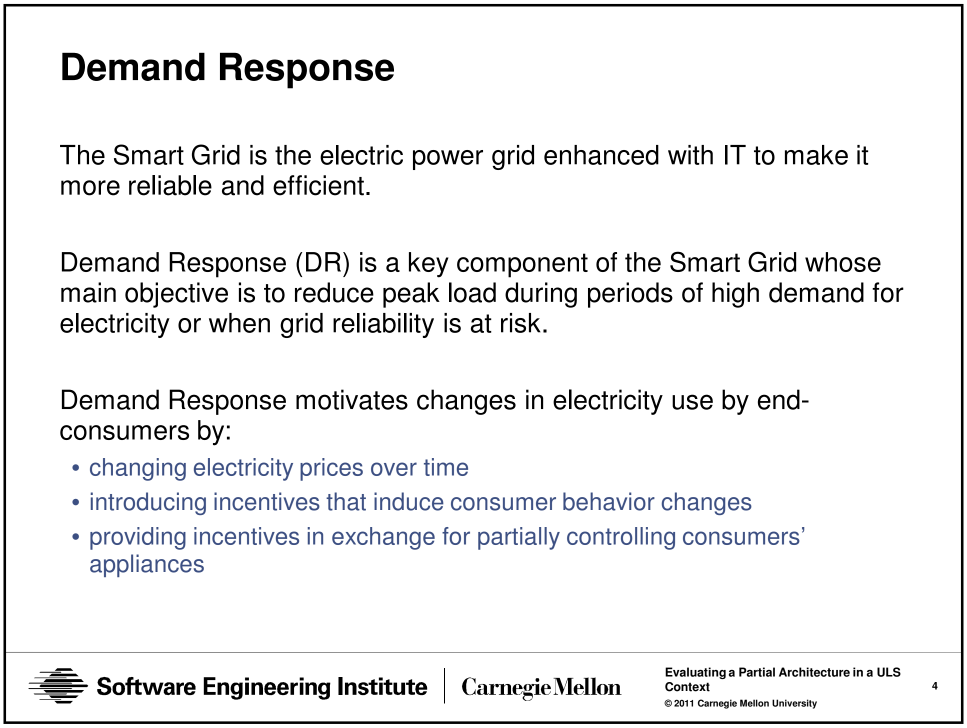
- The architecture may not yet exist for portions of the system.
- There may be competing architectures for the same functionality in different parts of the system.

There is not a concrete architecture to evaluate but an…

architecture landscape: a broad set of architectural decisions which represent a spectrum of potential architectures.

Analyzing an architecture landscape is beneficial in situations where there are

- many architectural decisions with far reaching consequences to be made
- many stakeholders
- many similar systems to be built



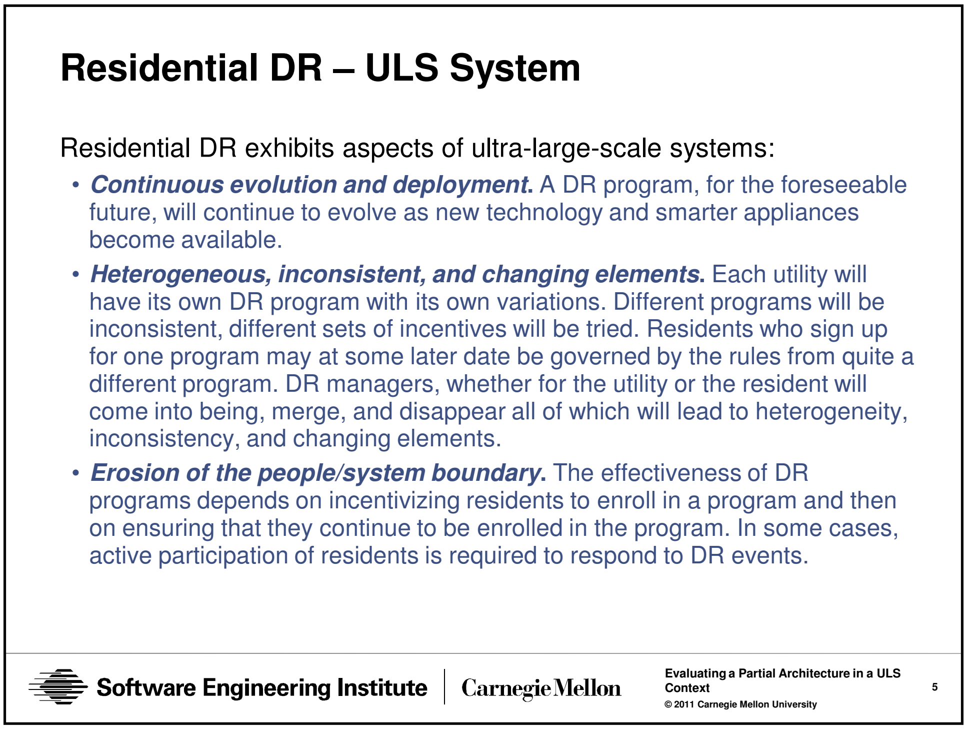
Demand Response

The Smart Grid is the electric power grid enhanced with IT to make it more reliable and efficient.

Demand Response (DR) is a key component of the Smart Grid whose main objective is to reduce peak load during periods of high demand for electricity or when grid reliability is at risk.

Demand Response motivates changes in electricity use by end-consumers by:

- changing electricity prices over time
- introducing incentives that induce consumer behavior changes
- providing incentives in exchange for partially controlling consumers' appliances



Residential DR – ULS System

Residential DR exhibits aspects of ultra-large-scale systems:

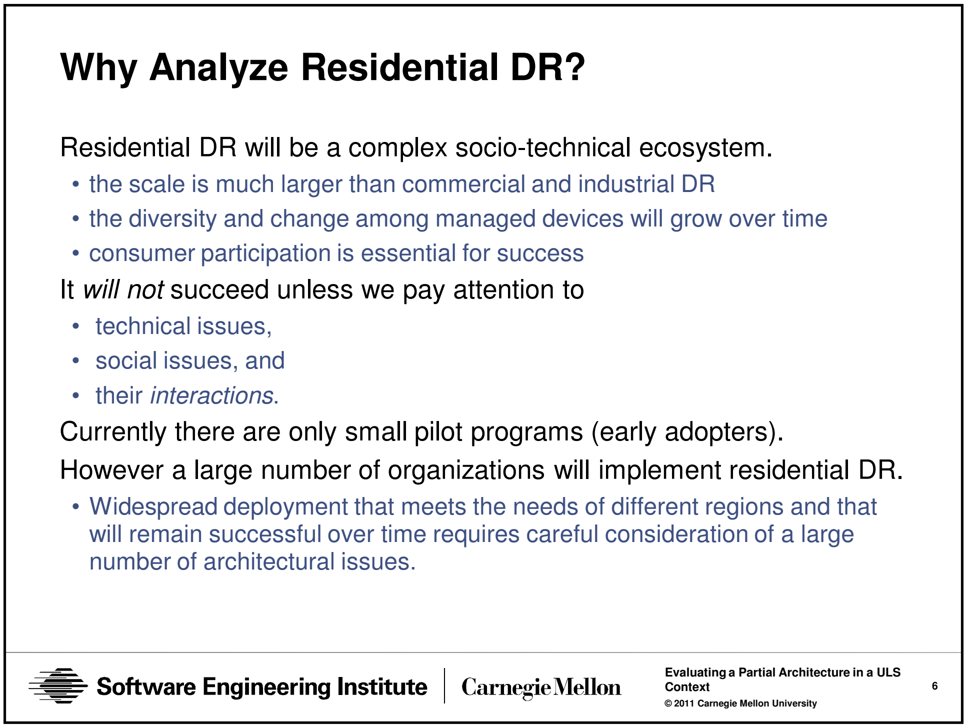
- ***Continuous evolution and deployment.*** A DR program, for the foreseeable future, will continue to evolve as new technology and smarter appliances become available.
- ***Heterogeneous, inconsistent, and changing elements.*** Each utility will have its own DR program with its own variations. Different programs will be inconsistent, different sets of incentives will be tried. Residents who sign up for one program may at some later date be governed by the rules from quite a different program. DR managers, whether for the utility or the resident will come into being, merge, and disappear all of which will lead to heterogeneity, inconsistency, and changing elements.
- ***Erosion of the people/system boundary.*** The effectiveness of DR programs depends on incentivizing residents to enroll in a program and then on ensuring that they continue to be enrolled in the program. In some cases, active participation of residents is required to respond to DR events.

Software Engineering Institute | Carnegie Mellon

Evaluating a Partial Architecture in a ULS Context

© 2011 Carnegie Mellon University

5



Why Analyze Residential DR?

Residential DR will be a complex socio-technical ecosystem.

- the scale is much larger than commercial and industrial DR
- the diversity and change among managed devices will grow over time
- consumer participation is essential for success

It *will not* succeed unless we pay attention to

- technical issues,
- social issues, and
- their *interactions*.

Currently there are only small pilot programs (early adopters).

However a large number of organizations will implement residential DR.

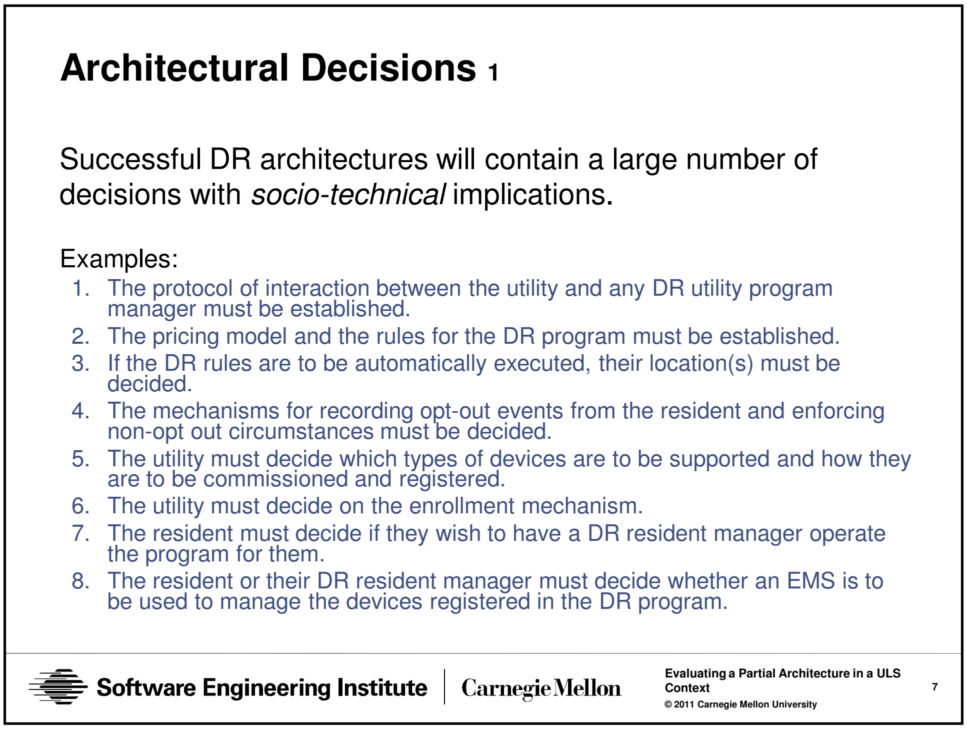
- Widespread deployment that meets the needs of different regions and that will remain successful over time requires careful consideration of a large number of architectural issues.

Software Engineering Institute | Carnegie Mellon

Evaluating a Partial Architecture in a ULS Context

© 2011 Carnegie Mellon University

6

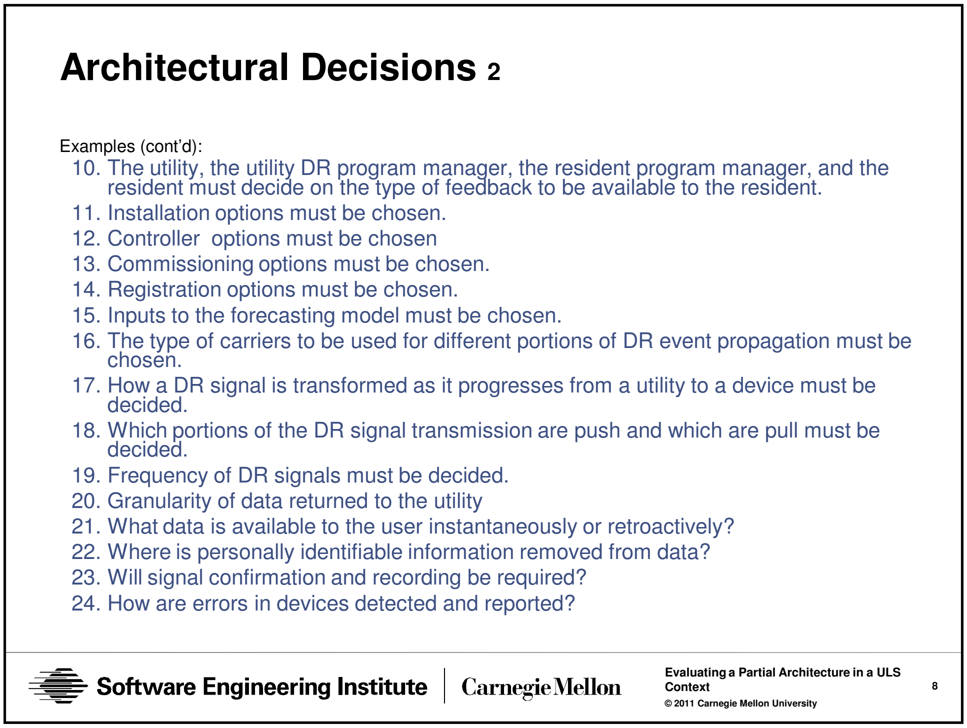


Architectural Decisions 1

Successful DR architectures will contain a large number of decisions with *socio-technical* implications.

Examples:

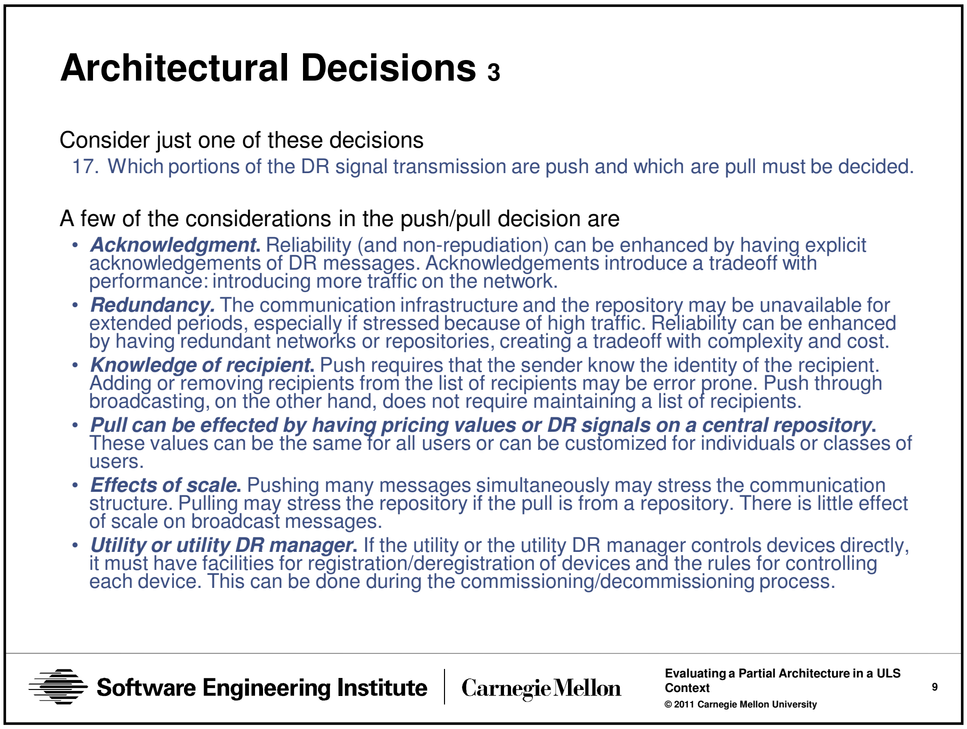
1. The protocol of interaction between the utility and any DR utility program manager must be established.
2. The pricing model and the rules for the DR program must be established.
3. If the DR rules are to be automatically executed, their location(s) must be decided.
4. The mechanisms for recording opt-out events from the resident and enforcing non-opt out circumstances must be decided.
5. The utility must decide which types of devices are to be supported and how they are to be commissioned and registered.
6. The utility must decide on the enrollment mechanism.
7. The resident must decide if they wish to have a DR resident manager operate the program for them.
8. The resident or their DR resident manager must decide whether an EMS is to be used to manage the devices registered in the DR program.



Architectural Decisions 2

Examples (cont'd):

10. The utility, the utility DR program manager, the resident program manager, and the resident must decide on the type of feedback to be available to the resident.
11. Installation options must be chosen.
12. Controller options must be chosen
13. Commissioning options must be chosen.
14. Registration options must be chosen.
15. Inputs to the forecasting model must be chosen.
16. The type of carriers to be used for different portions of DR event propagation must be chosen.
17. How a DR signal is transformed as it progresses from a utility to a device must be decided.
18. Which portions of the DR signal transmission are push and which are pull must be decided.
19. Frequency of DR signals must be decided.
20. Granularity of data returned to the utility
21. What data is available to the user instantaneously or retroactively?
22. Where is personally identifiable information removed from data?
23. Will signal confirmation and recording be required?
24. How are errors in devices detected and reported?



Architectural Decisions 3

Consider just one of these decisions

17. Which portions of the DR signal transmission are push and which are pull must be decided.

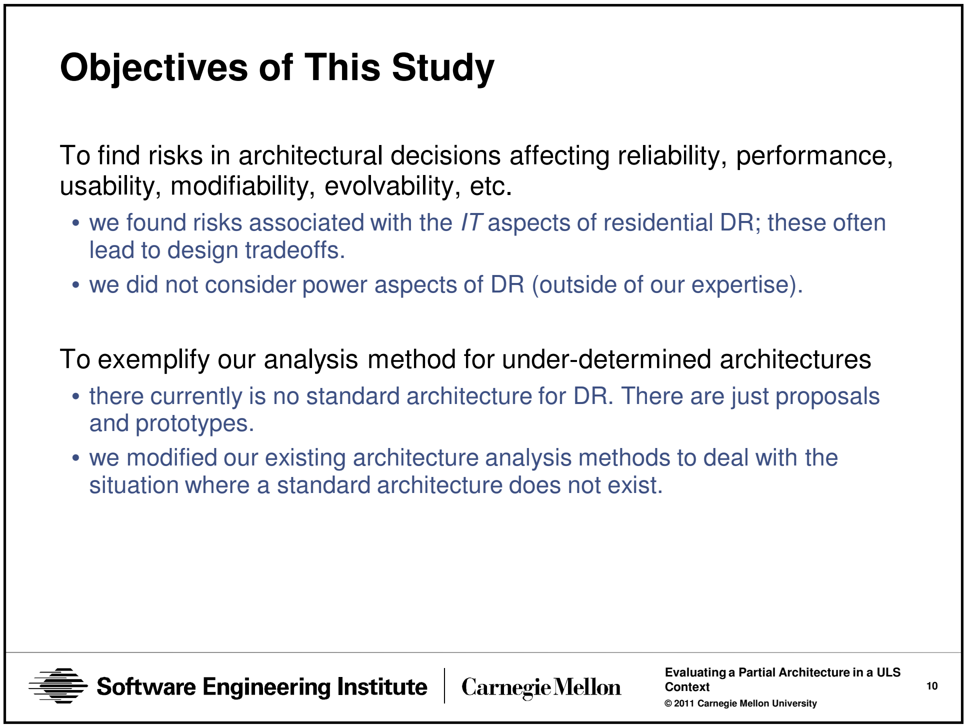
A few of the considerations in the push/pull decision are

- ***Acknowledgment.*** Reliability (and non-repudiation) can be enhanced by having explicit acknowledgements of DR messages. Acknowledgements introduce a tradeoff with performance: introducing more traffic on the network.
- ***Redundancy.*** The communication infrastructure and the repository may be unavailable for extended periods, especially if stressed because of high traffic. Reliability can be enhanced by having redundant networks or repositories, creating a tradeoff with complexity and cost.
- ***Knowledge of recipient.*** Push requires that the sender know the identity of the recipient. Adding or removing recipients from the list of recipients may be error prone. Push through broadcasting, on the other hand, does not require maintaining a list of recipients.
- ***Pull can be effected by having pricing values or DR signals on a central repository.*** These values can be the same for all users or can be customized for individuals or classes of users.
- ***Effects of scale.*** Pushing many messages simultaneously may stress the communication structure. Pulling may stress the repository if the pull is from a repository. There is little effect of scale on broadcast messages.
- ***Utility or utility DR manager.*** If the utility or the utility DR manager controls devices directly, it must have facilities for registration/deregistration of devices and the rules for controlling each device. This can be done during the commissioning/decommissioning process.

Software Engineering Institute | Carnegie Mellon

Evaluating a Partial Architecture in a ULS Context 9

© 2011 Carnegie Mellon University



Objectives of This Study

To find risks in architectural decisions affecting reliability, performance, usability, modifiability, evolvability, etc.

- we found risks associated with the *IT* aspects of residential DR; these often lead to design tradeoffs.
- we did not consider power aspects of DR (outside of our expertise).

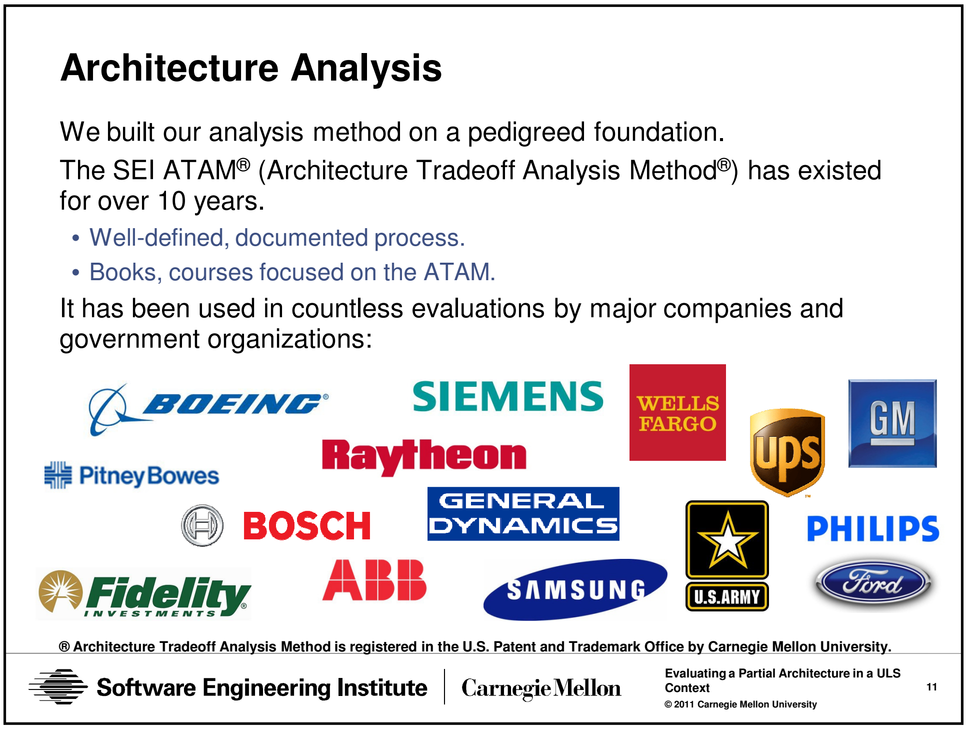
To exemplify our analysis method for under-determined architectures

- there currently is no standard architecture for DR. There are just proposals and prototypes.
- we modified our existing architecture analysis methods to deal with the situation where a standard architecture does not exist.

Software Engineering Institute | Carnegie Mellon

Evaluating a Partial Architecture in a ULS Context 10

© 2011 Carnegie Mellon University



Architecture Analysis

We built our analysis method on a pedigreed foundation.

The SEI ATAM® (Architecture Tradeoff Analysis Method®) has existed for over 10 years.

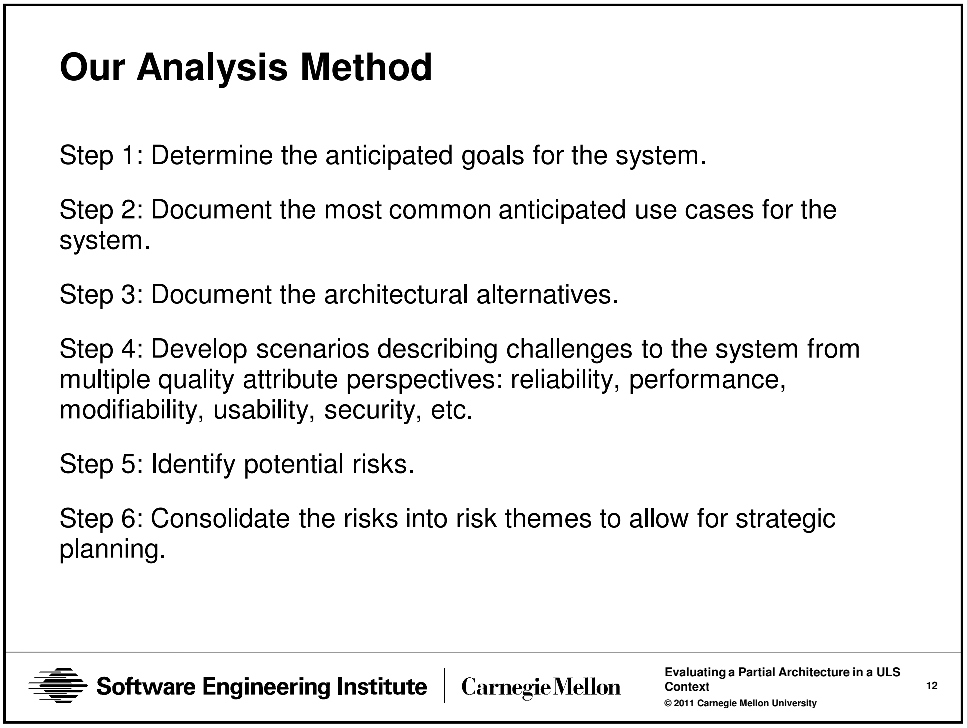
- Well-defined, documented process.
- Books, courses focused on the ATAM.

It has been used in countless evaluations by major companies and government organizations:

® Architecture Tradeoff Analysis Method is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

Software Engineering Institute | Carnegie Mellon

Evaluating a Partial Architecture in a ULS Context 11



Our Analysis Method

Step 1: Determine the anticipated goals for the system.

Step 2: Document the most common anticipated use cases for the system.

Step 3: Document the architectural alternatives.

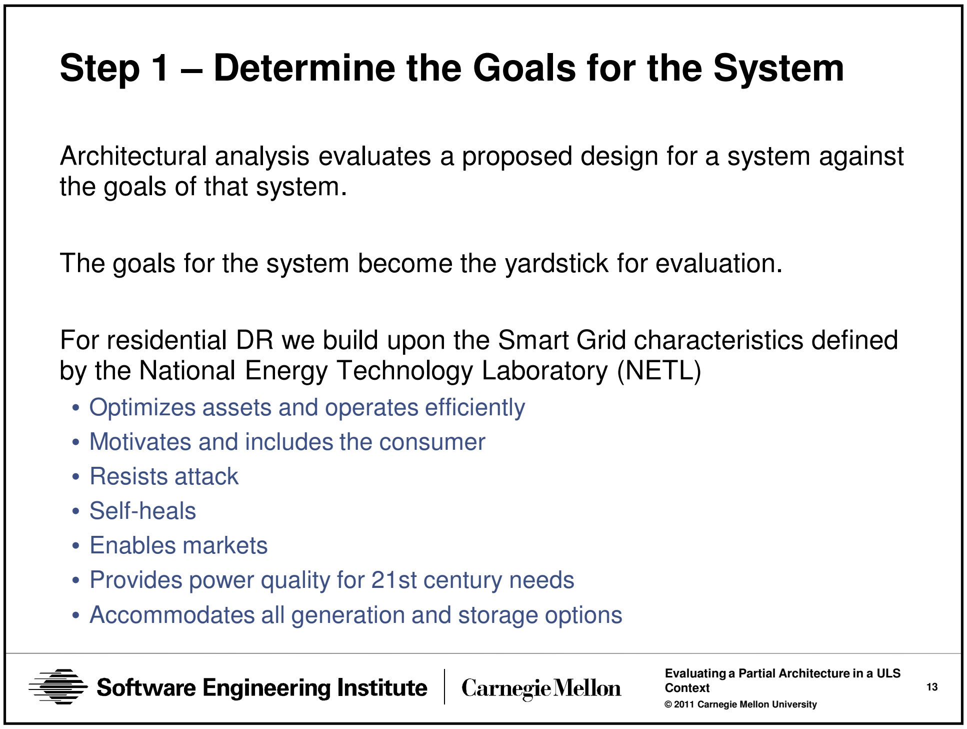
Step 4: Develop scenarios describing challenges to the system from multiple quality attribute perspectives: reliability, performance, modifiability, usability, security, etc.

Step 5: Identify potential risks.

Step 6: Consolidate the risks into risk themes to allow for strategic planning.

Software Engineering Institute | Carnegie Mellon

Evaluating a Partial Architecture in a ULS Context 12



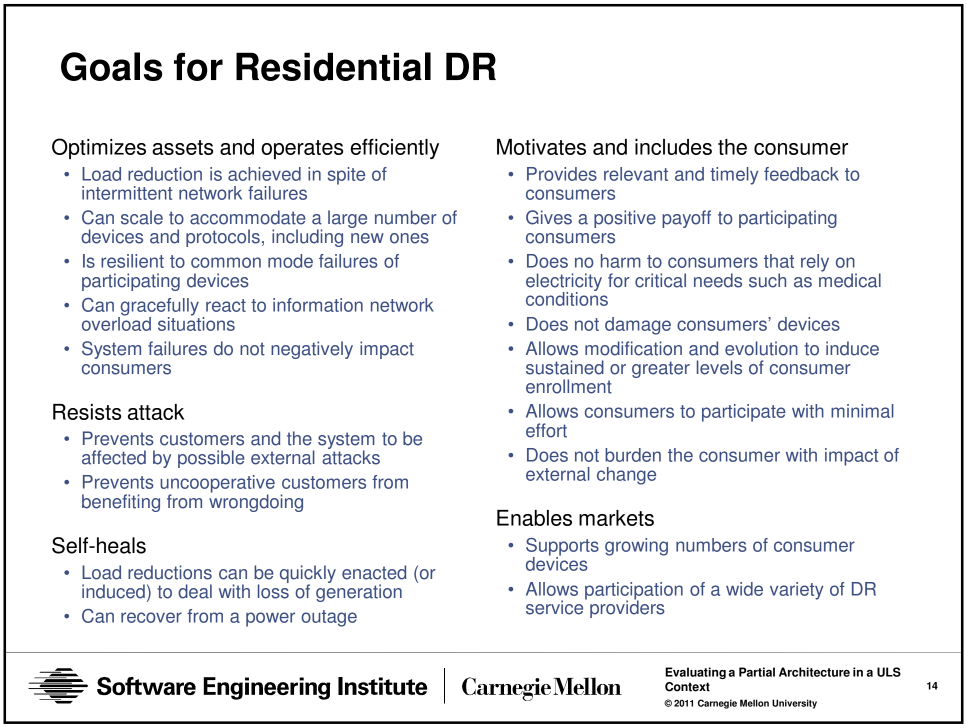
Step 1 – Determine the Goals for the System

Architectural analysis evaluates a proposed design for a system against the goals of that system.

The goals for the system become the yardstick for evaluation.

For residential DR we build upon the Smart Grid characteristics defined by the National Energy Technology Laboratory (NETL)

- Optimizes assets and operates efficiently
- Motivates and includes the consumer
- Resists attack
- Self-heals
- Enables markets
- Provides power quality for 21st century needs
- Accommodates all generation and storage options



Goals for Residential DR

Optimizes assets and operates efficiently

- Load reduction is achieved in spite of intermittent network failures
- Can scale to accommodate a large number of devices and protocols, including new ones
- Is resilient to common mode failures of participating devices
- Can gracefully react to information network overload situations
- System failures do not negatively impact consumers

Resists attack

- Prevents customers and the system to be affected by possible external attacks
- Prevents uncooperative customers from benefiting from wrongdoing

Self-heals

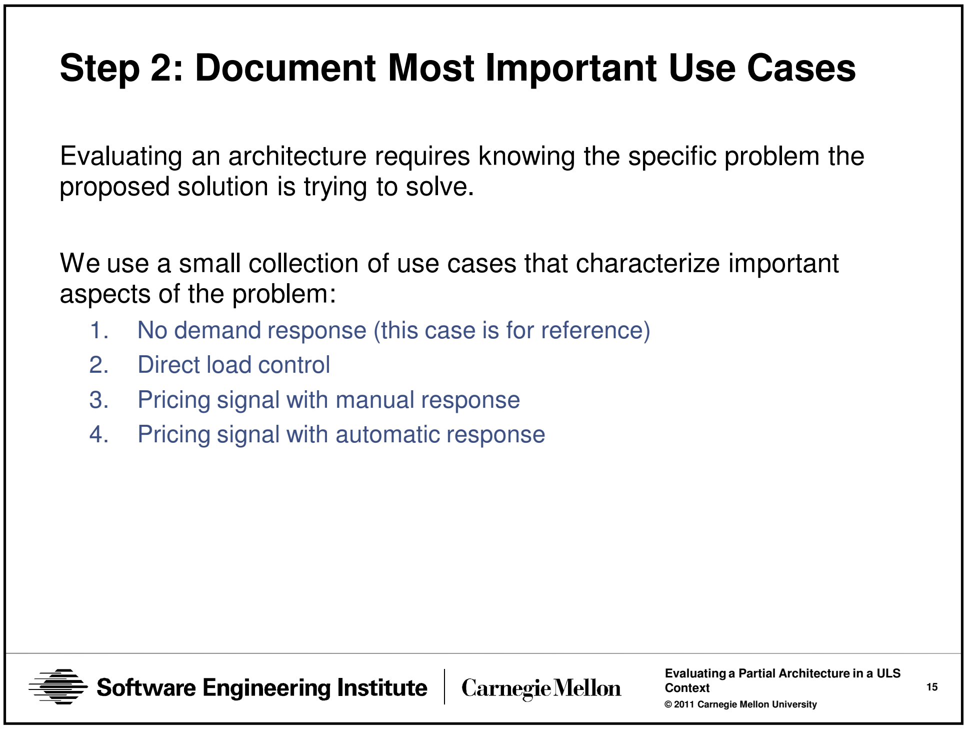
- Load reductions can be quickly enacted (or induced) to deal with loss of generation
- Can recover from a power outage

Motivates and includes the consumer

- Provides relevant and timely feedback to consumers
- Gives a positive payoff to participating consumers
- Does no harm to consumers that rely on electricity for critical needs such as medical conditions
- Does not damage consumers' devices
- Allows modification and evolution to induce sustained or greater levels of consumer enrollment
- Allows consumers to participate with minimal effort
- Does not burden the consumer with impact of external change

Enables markets

- Supports growing numbers of consumer devices
- Allows participation of a wide variety of DR service providers

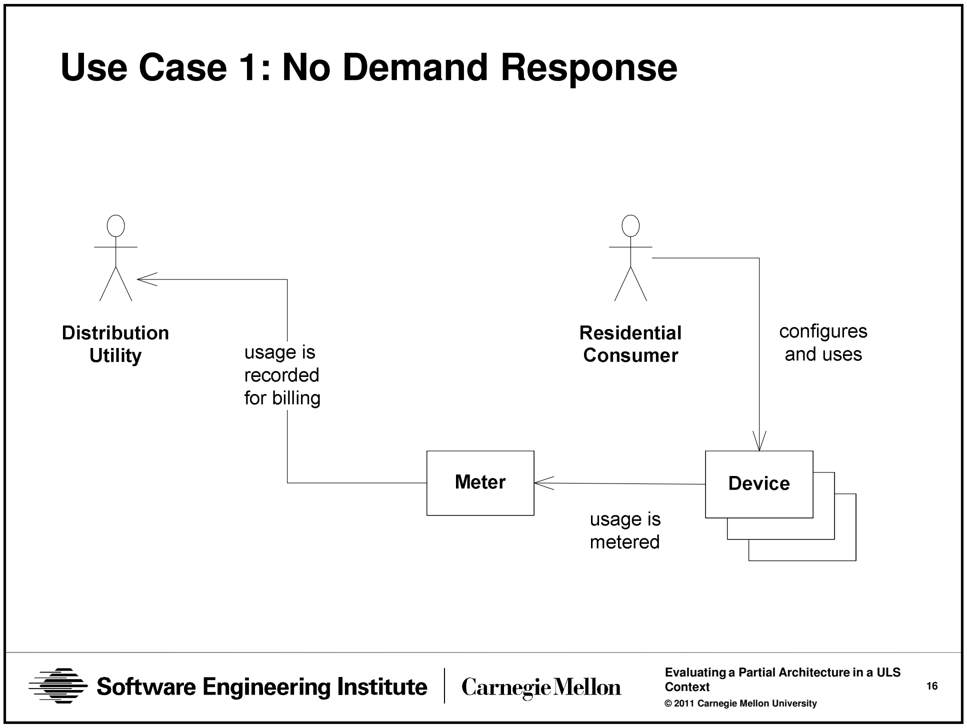


Step 2: Document Most Important Use Cases

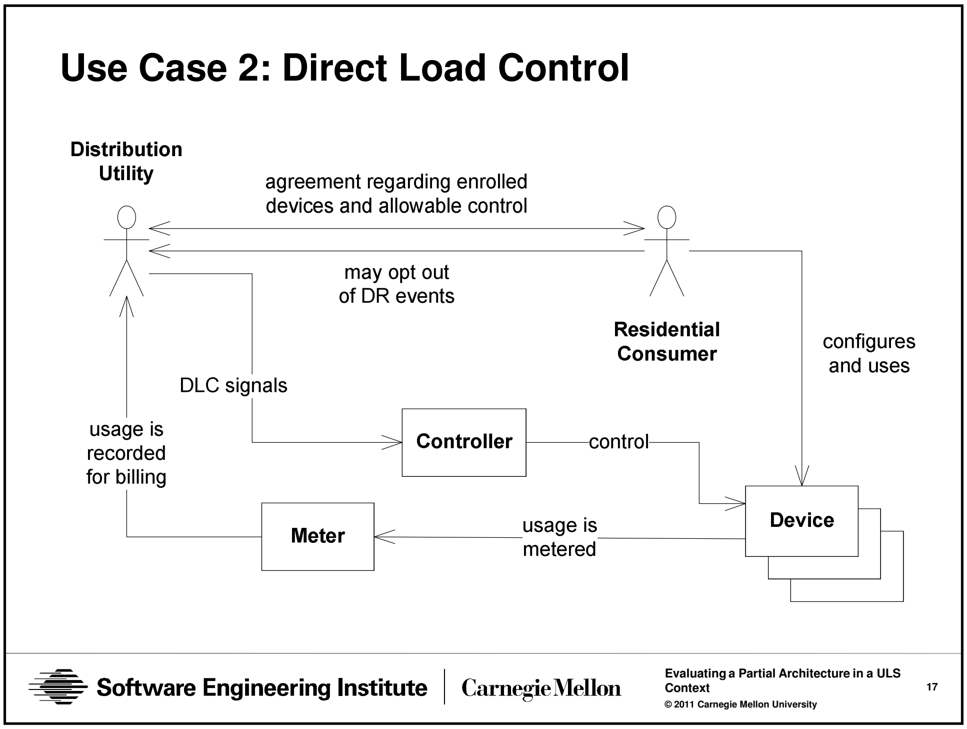
Evaluating an architecture requires knowing the specific problem the proposed solution is trying to solve.

We use a small collection of use cases that characterize important aspects of the problem:

1. No demand response (this case is for reference)
2. Direct load control
3. Pricing signal with manual response
4. Pricing signal with automatic response



Use Case 1: No Demand Response



Use Case 2: Direct Load Control

Distribution Utility

agreement regarding enrolled devices and allowable control

may opt out of DR events

Residential Consumer

configures and uses

DLC signals

usage is recorded for billing

Controller

control

Device

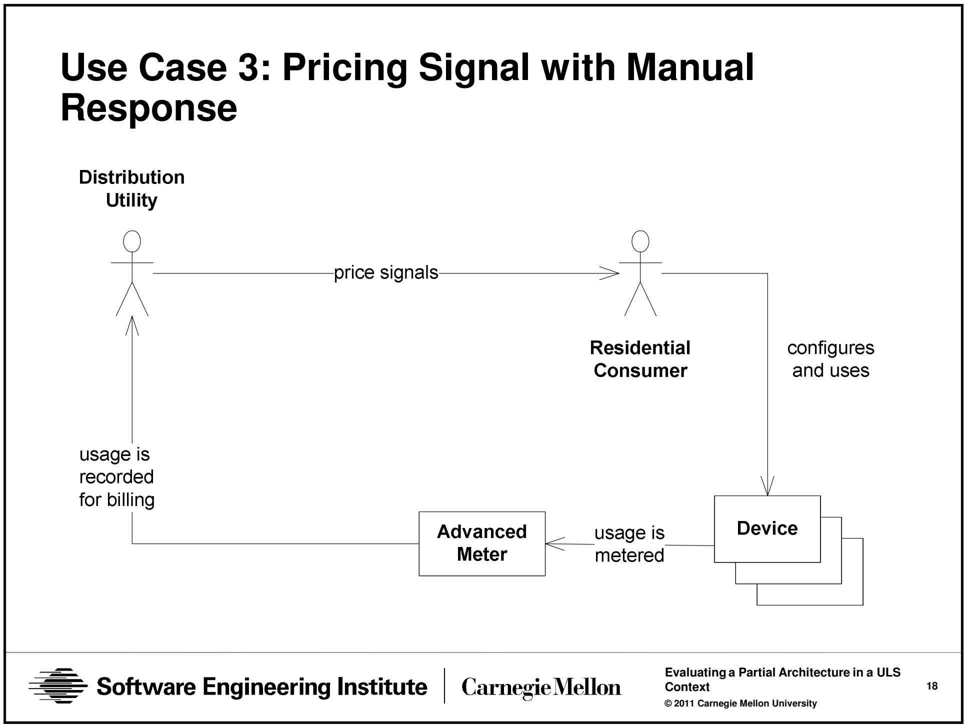
Meter

usage is metered

Software Engineering Institute | Carnegie Mellon

Evaluating a Partial Architecture in a ULS Context 17

© 2011 Carnegie Mellon University



Use Case 3: Pricing Signal with Manual Response

Distribution Utility

price signals

Residential Consumer

configures and uses

usage is recorded for billing

Advanced Meter

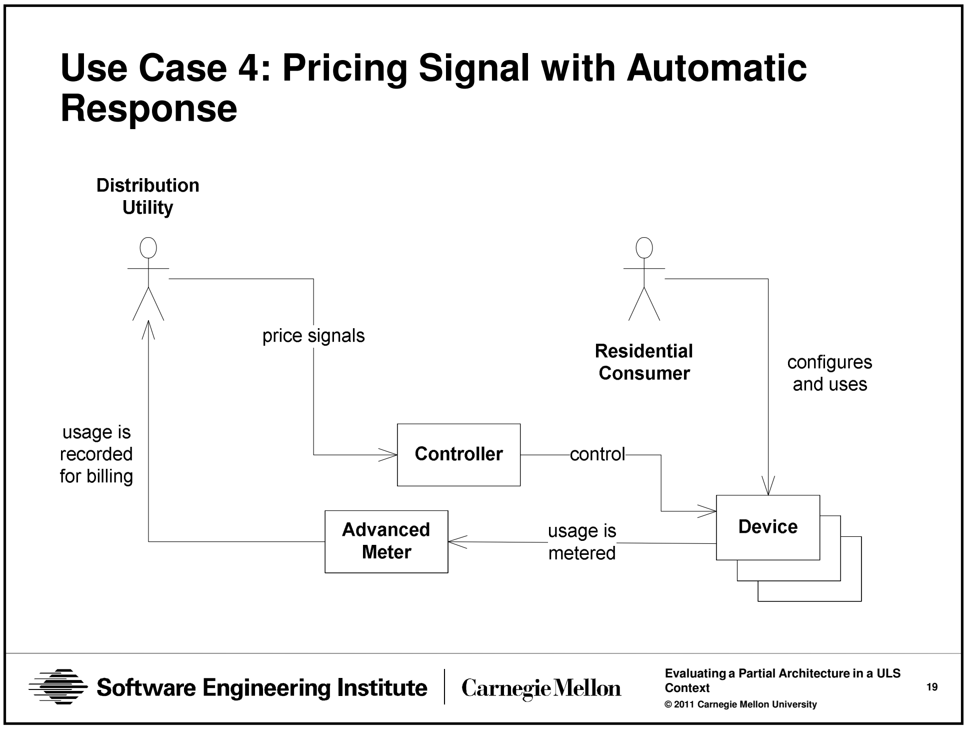
usage is metered

Device

Software Engineering Institute | Carnegie Mellon

Evaluating a Partial Architecture in a ULS Context 18

© 2011 Carnegie Mellon University



Use Case 4: Pricing Signal with Automatic Response

Distribution Utility

price signals

Residential Consumer

configures and uses

usage is recorded for billing

Controller

control

Device

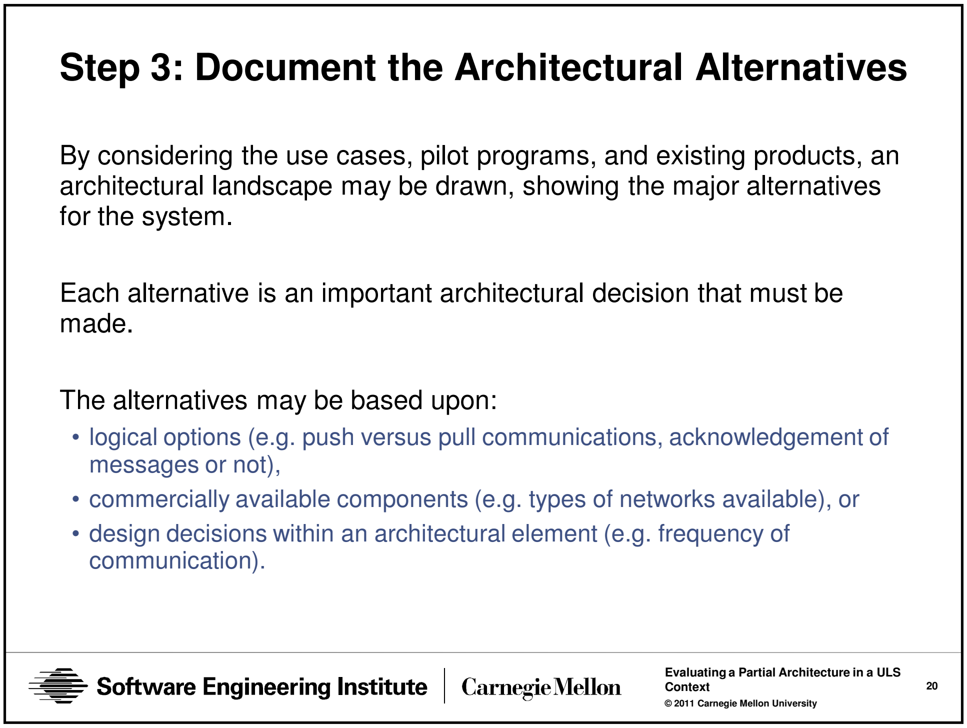
Advanced Meter

usage is metered

Software Engineering Institute | Carnegie Mellon

Evaluating a Partial Architecture in a ULS Context 19

© 2011 Carnegie Mellon University



Step 3: Document the Architectural Alternatives

By considering the use cases, pilot programs, and existing products, an architectural landscape may be drawn, showing the major alternatives for the system.

Each alternative is an important architectural decision that must be made.

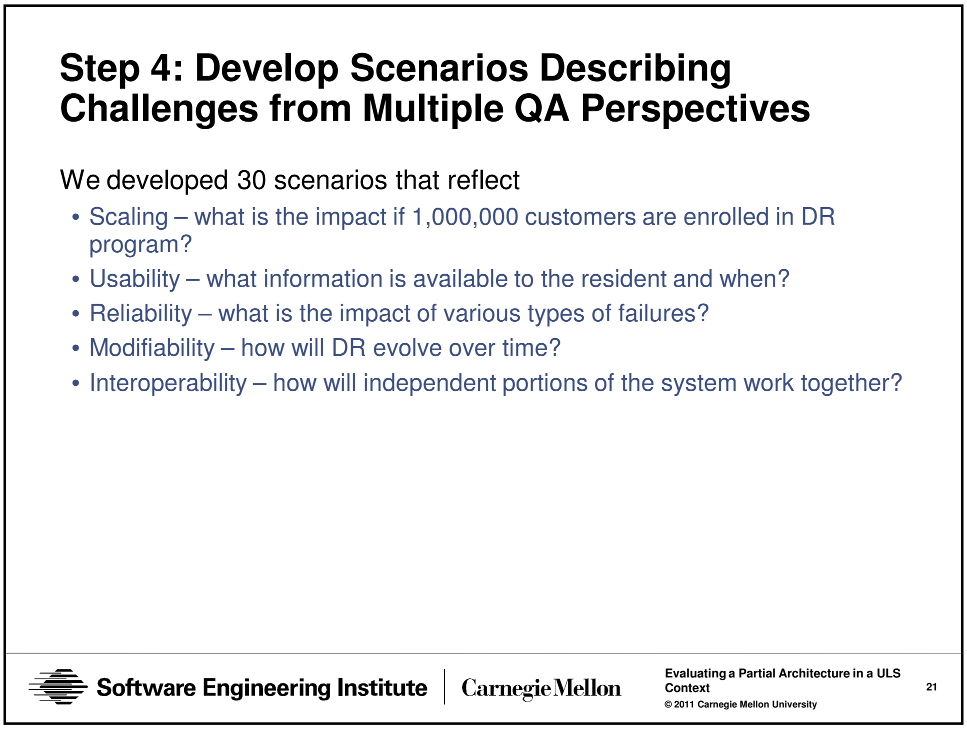
The alternatives may be based upon:

- logical options (e.g. push versus pull communications, acknowledgement of messages or not),
- commercially available components (e.g. types of networks available), or
- design decisions within an architectural element (e.g. frequency of communication).

Software Engineering Institute | Carnegie Mellon

Evaluating a Partial Architecture in a ULS Context 20

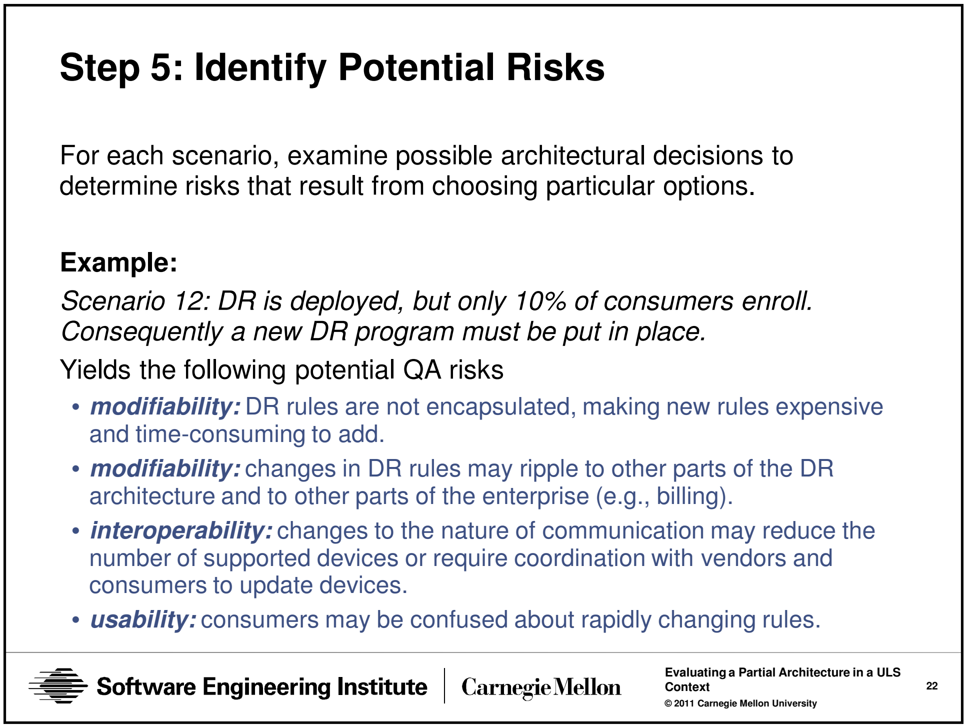
© 2011 Carnegie Mellon University



Step 4: Develop Scenarios Describing Challenges from Multiple QA Perspectives

We developed 30 scenarios that reflect

- Scaling – what is the impact if 1,000,000 customers are enrolled in DR program?
- Usability – what information is available to the resident and when?
- Reliability – what is the impact of various types of failures?
- Modifiability – how will DR evolve over time?
- Interoperability – how will independent portions of the system work together?



Step 5: Identify Potential Risks

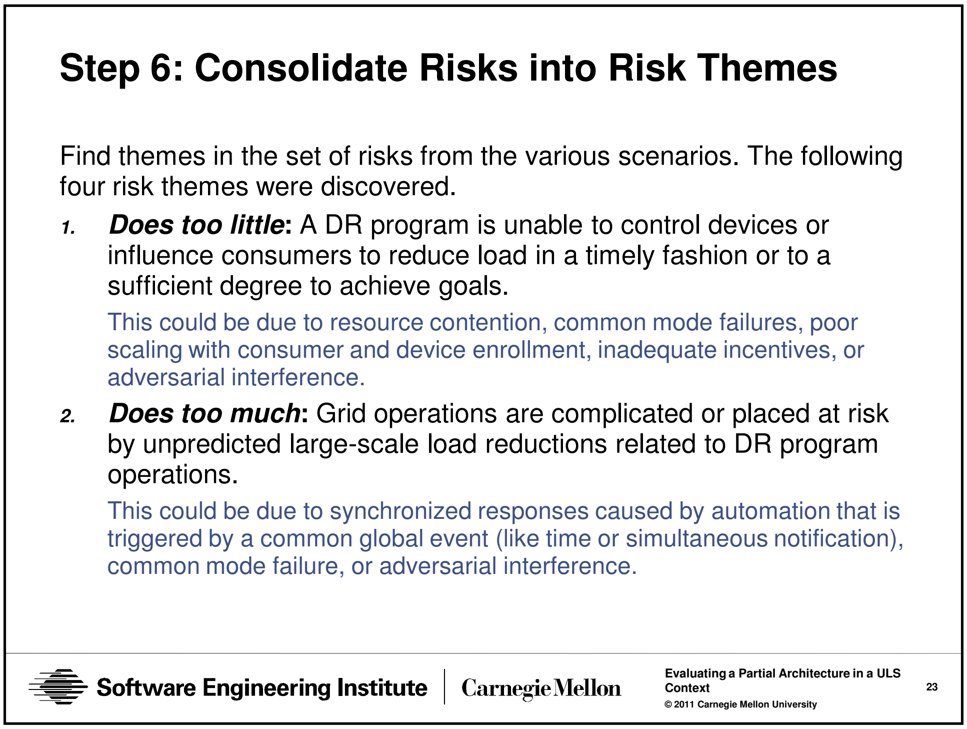
For each scenario, examine possible architectural decisions to determine risks that result from choosing particular options.

Example:

Scenario 12: DR is deployed, but only 10% of consumers enroll. Consequently a new DR program must be put in place.

Yields the following potential QA risks

- ***modifiability:*** DR rules are not encapsulated, making new rules expensive and time-consuming to add.
- ***modifiability:*** changes in DR rules may ripple to other parts of the DR architecture and to other parts of the enterprise (e.g., billing).
- ***interoperability:*** changes to the nature of communication may reduce the number of supported devices or require coordination with vendors and consumers to update devices.
- ***usability:*** consumers may be confused about rapidly changing rules.



Step 6: Consolidate Risks into Risk Themes

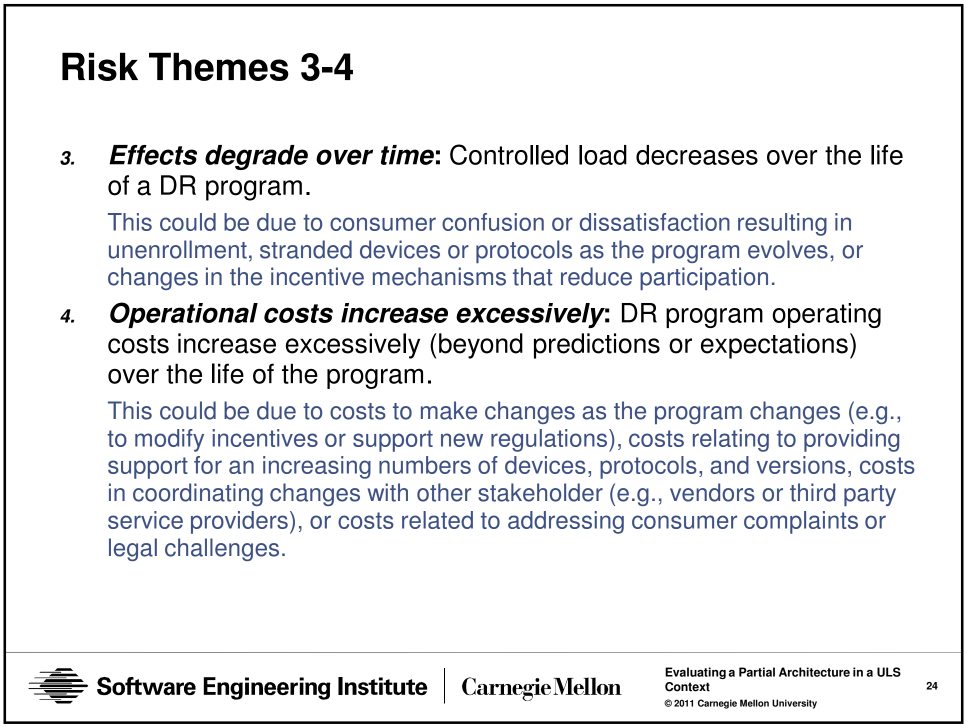
Find themes in the set of risks from the various scenarios. The following four risk themes were discovered.

1. ***Does too little***: A DR program is unable to control devices or influence consumers to reduce load in a timely fashion or to a sufficient degree to achieve goals.

   This could be due to resource contention, common mode failures, poor scaling with consumer and device enrollment, inadequate incentives, or adversarial interference.

2. ***Does too much***: Grid operations are complicated or placed at risk by unpredicted large-scale load reductions related to DR program operations.

   This could be due to synchronized responses caused by automation that is triggered by a common global event (like time or simultaneous notification), common mode failure, or adversarial interference.



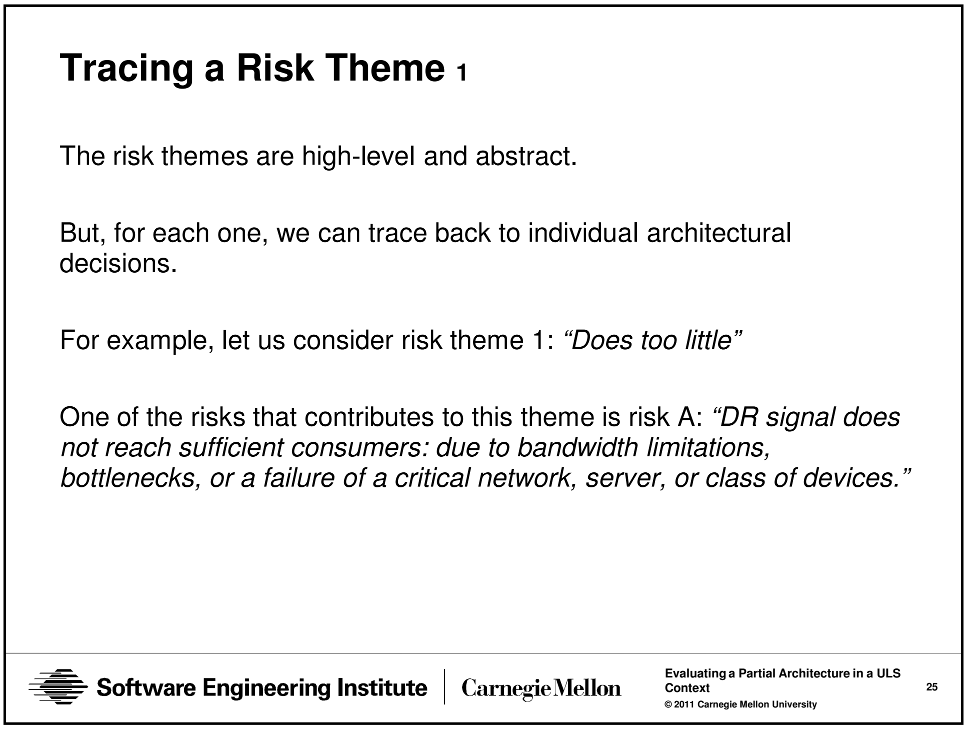
Risk Themes 3-4

3. ***Effects degrade over time***: Controlled load decreases over the life of a DR program.

   This could be due to consumer confusion or dissatisfaction resulting in unenrollment, stranded devices or protocols as the program evolves, or changes in the incentive mechanisms that reduce participation.

4. ***Operational costs increase excessively***: DR program operating costs increase excessively (beyond predictions or expectations) over the life of the program.

   This could be due to costs to make changes as the program changes (e.g., to modify incentives or support new regulations), costs relating to providing support for an increasing numbers of devices, protocols, and versions, costs in coordinating changes with other stakeholder (e.g., vendors or third party service providers), or costs related to addressing consumer complaints or legal challenges.



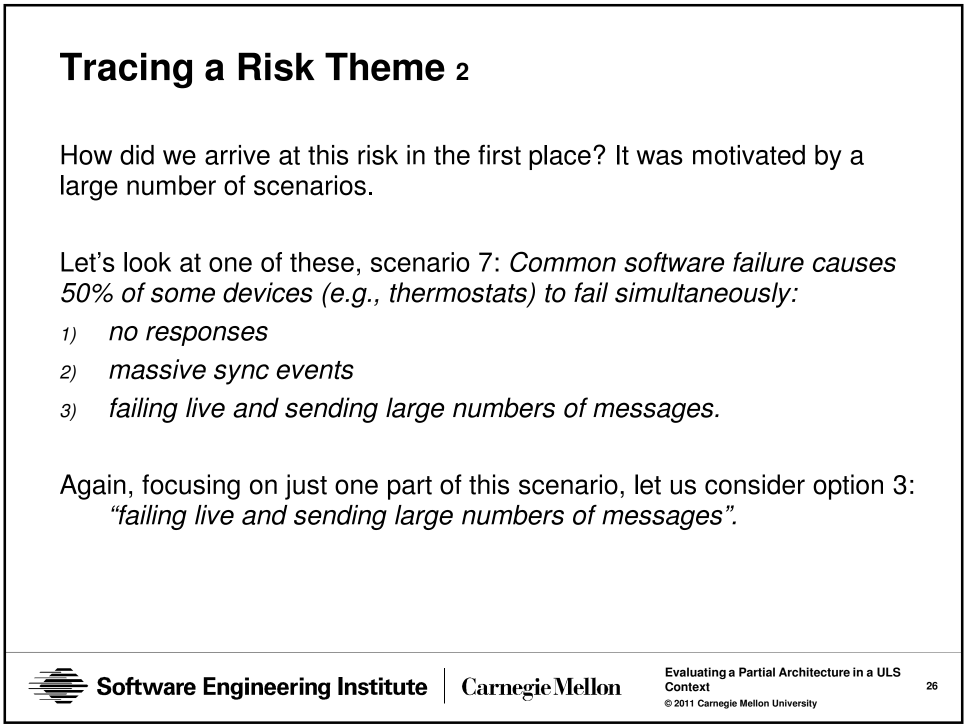
Tracing a Risk Theme 1

The risk themes are high-level and abstract.

But, for each one, we can trace back to individual architectural decisions.

For example, let us consider risk theme 1: *“Does too little”*

One of the risks that contributes to this theme is risk A: *“DR signal does not reach sufficient consumers: due to bandwidth limitations, bottlenecks, or a failure of a critical network, server, or class of devices.”*



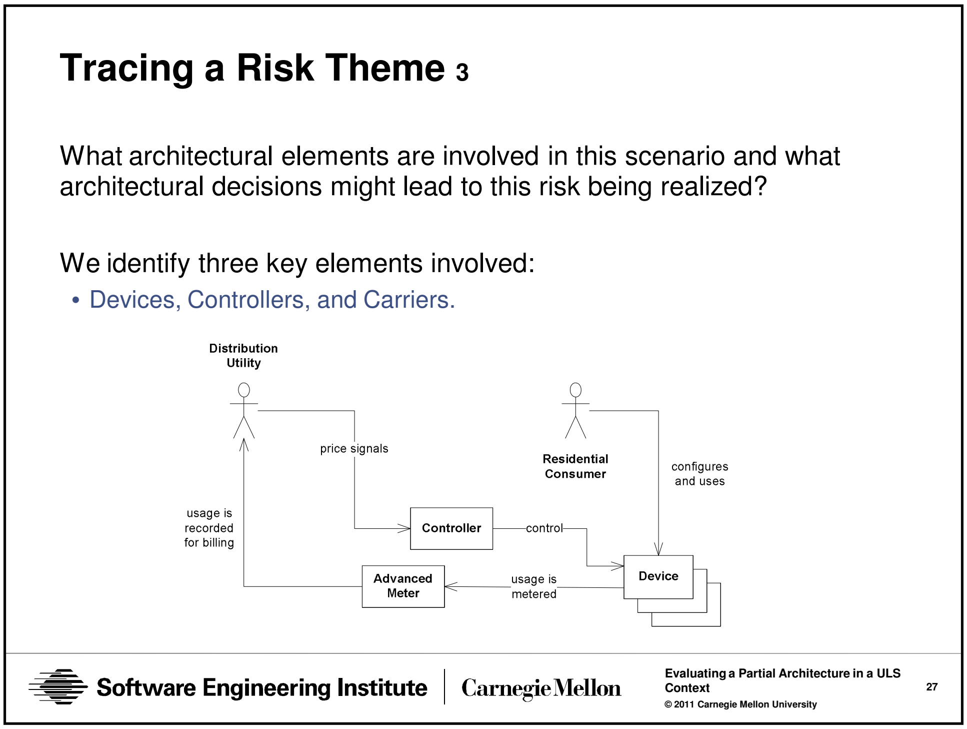
Tracing a Risk Theme 2

How did we arrive at this risk in the first place? It was motivated by a large number of scenarios.

Let’s look at one of these, scenario 7: *Common software failure causes 50% of some devices (e.g., thermostats) to fail simultaneously:*

1) *no responses*
2) *massive sync events*
3) *failing live and sending large numbers of messages.*

Again, focusing on just one part of this scenario, let us consider option 3: *“failing live and sending large numbers of messages”.*



Tracing a Risk Theme 3

What architectural elements are involved in this scenario and what architectural decisions might lead to this risk being realized?

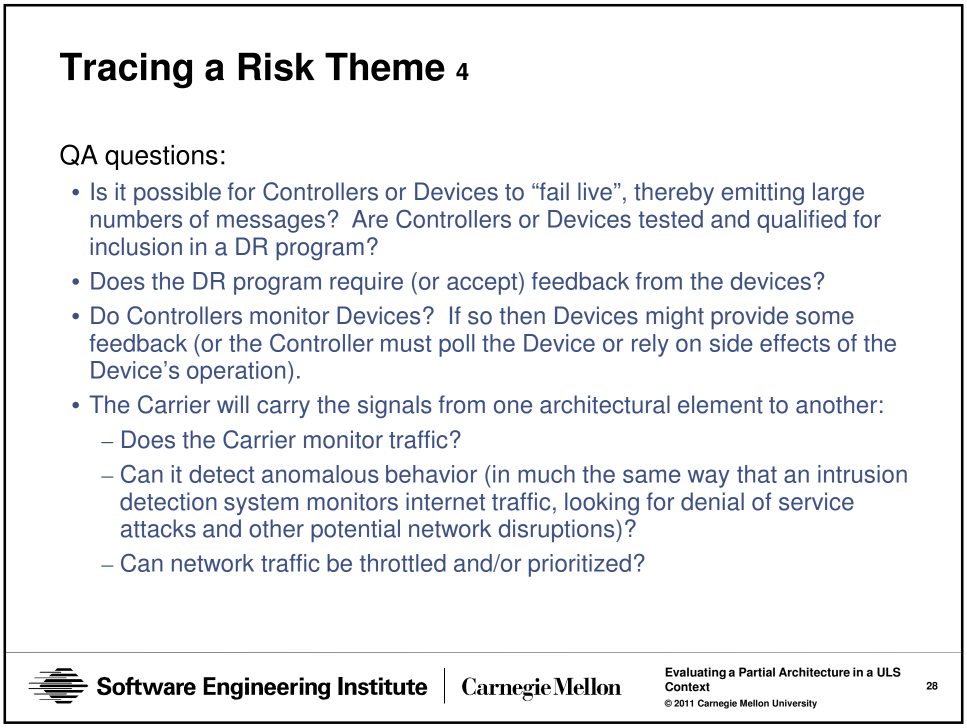
We identify three key elements involved:

- Devices, Controllers, and Carriers.

Software Engineering Institute | Carnegie Mellon

Evaluating a Partial Architecture in a ULS Context 27

© 2011 Carnegie Mellon University



Tracing a Risk Theme 4

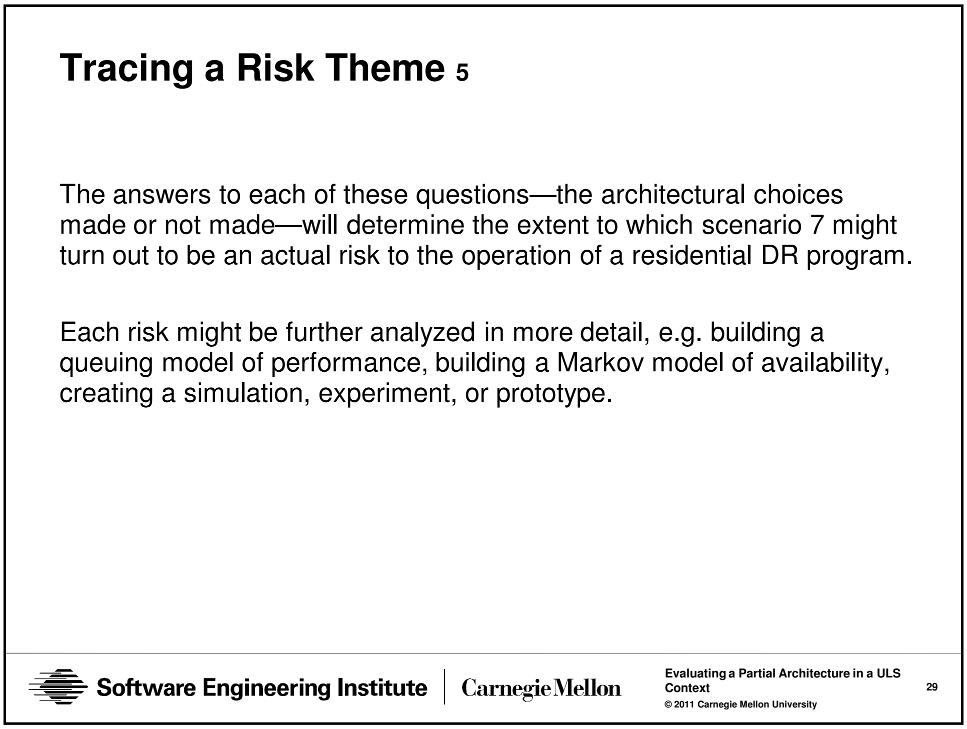
QA questions:

- Is it possible for Controllers or Devices to “fail live”, thereby emitting large numbers of messages? Are Controllers or Devices tested and qualified for inclusion in a DR program?
- Does the DR program require (or accept) feedback from the devices?
- Do Controllers monitor Devices? If so then Devices might provide some feedback (or the Controller must poll the Device or rely on side effects of the Device’s operation).
- The Carrier will carry the signals from one architectural element to another:
  - Does the Carrier monitor traffic?
  - Can it detect anomalous behavior (in much the same way that an intrusion detection system monitors internet traffic, looking for denial of service attacks and other potential network disruptions)?
  - Can network traffic be throttled and/or prioritized?

Software Engineering Institute | Carnegie Mellon

Evaluating a Partial Architecture in a ULS Context 28

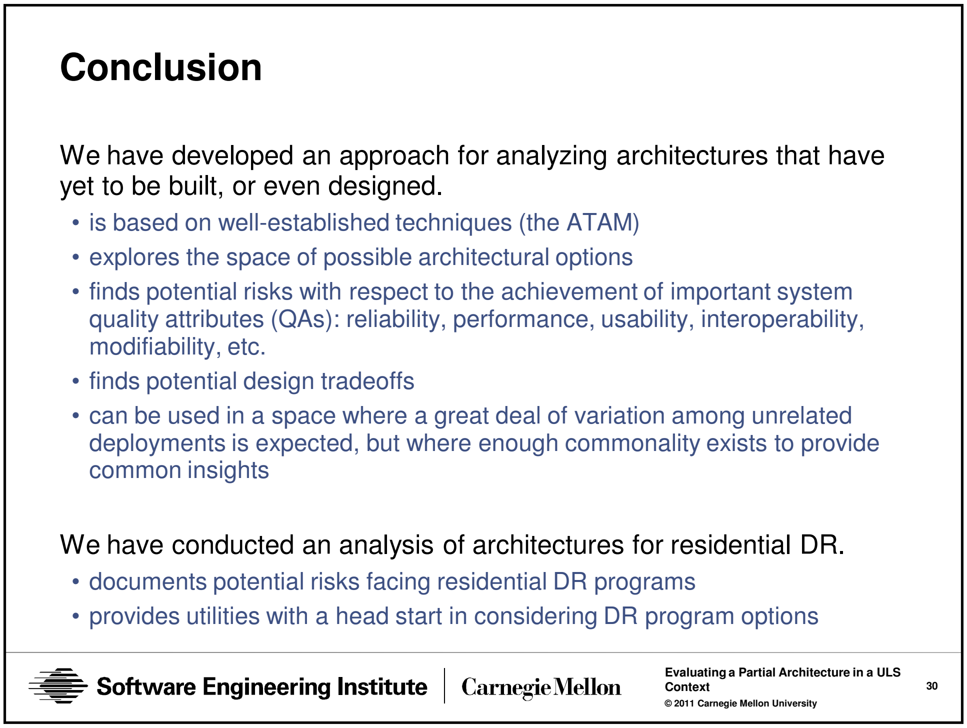
© 2011 Carnegie Mellon University



Tracing a Risk Theme 5

The answers to each of these questions—the architectural choices made or not made—will determine the extent to which scenario 7 might turn out to be an actual risk to the operation of a residential DR program.

Each risk might be further analyzed in more detail, e.g. building a queuing model of performance, building a Markov model of availability, creating a simulation, experiment, or prototype.



Conclusion

We have developed an approach for analyzing architectures that have yet to be built, or even designed.

- is based on well-established techniques (the ATAM)
- explores the space of possible architectural options
- finds potential risks with respect to the achievement of important system quality attributes (QAs): reliability, performance, usability, interoperability, modifiability, etc.
- finds potential design tradeoffs
- can be used in a space where a great deal of variation among unrelated deployments is expected, but where enough commonality exists to provide common insights

We have conducted an analysis of architectures for residential DR.

- documents potential risks facing residential DR programs
- provides utilities with a head start in considering DR program options