pier

What is System Integration?

Grant Gilchrist

EnerNex Corporation

November 29, 2005

Agenda

- What we mean by “integration”
- How it’s usually done
- How to do it right

Possible Utility Programs Involving Consumer Sites

- Automated Meter Reading
- Application of advanced tariffs
- Load control in response to pricing
- Load curtailment in emergencies
- Load control for grid optimization
- Load limiting due to lack of payment
- Outage management
- Theft detection
- Remote connect/disconnect, customer management
- Bill pre-payment

- Customer viewing of usage data
- Building management systems
- Distributed generation
- Bidding into demand and generation markets
- Power quality monitoring/control
- Minimizing system losses through consumer Volt/VAR control
- Non-energy applications: emergency notification, security, medical, entertainment

It doesn't make sense to design them all separately!

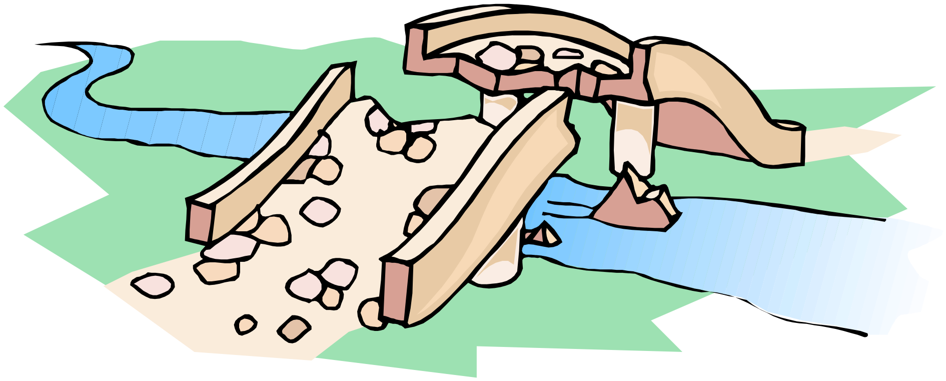
DANGER: Integration without Architecture

- **It is not sufficient just to integrate systems haphazardly**
- **There must be an overarching plan, or blueprint**
- **Otherwise, some vital functions may be prevented because of:**
  - High costs
  - Missing interfaces
  - Missing products
  - Missing technology

Impact of Poor Integration

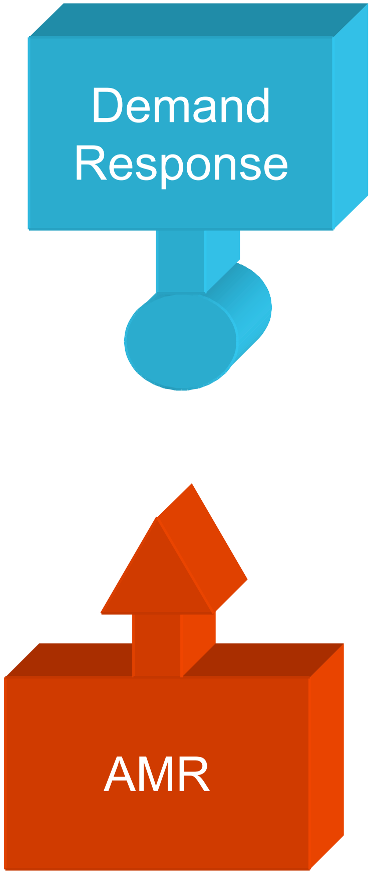
- Resources squandered on unimportant features
- Incomprehensible user interfaces
- Incompatible products
- Missing features
- Unsafe user environments
- Un-testable features
- Rework (with associated cost) when errors are discovered

Can lead to
"You can't get there from here"



HOW IT'S USUALLY DONE

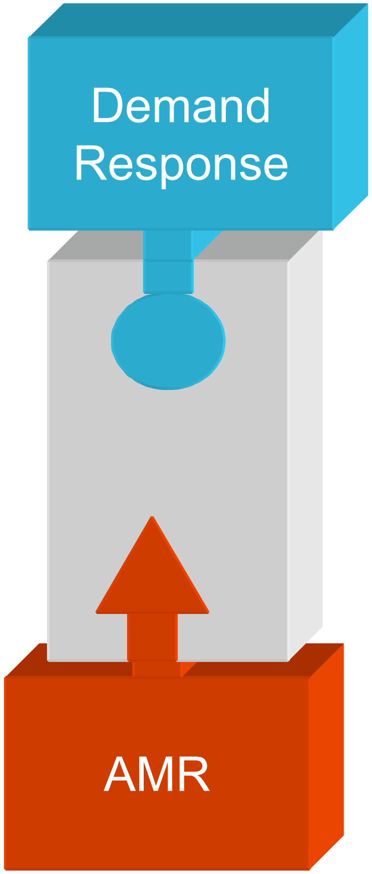
Building Isolated Systems



- Utilities tend to develop systems in isolation
- For example, AMR and demand response
- Neither project is typically developed with the other in mind.

HOW IT’S USUALLY DONE

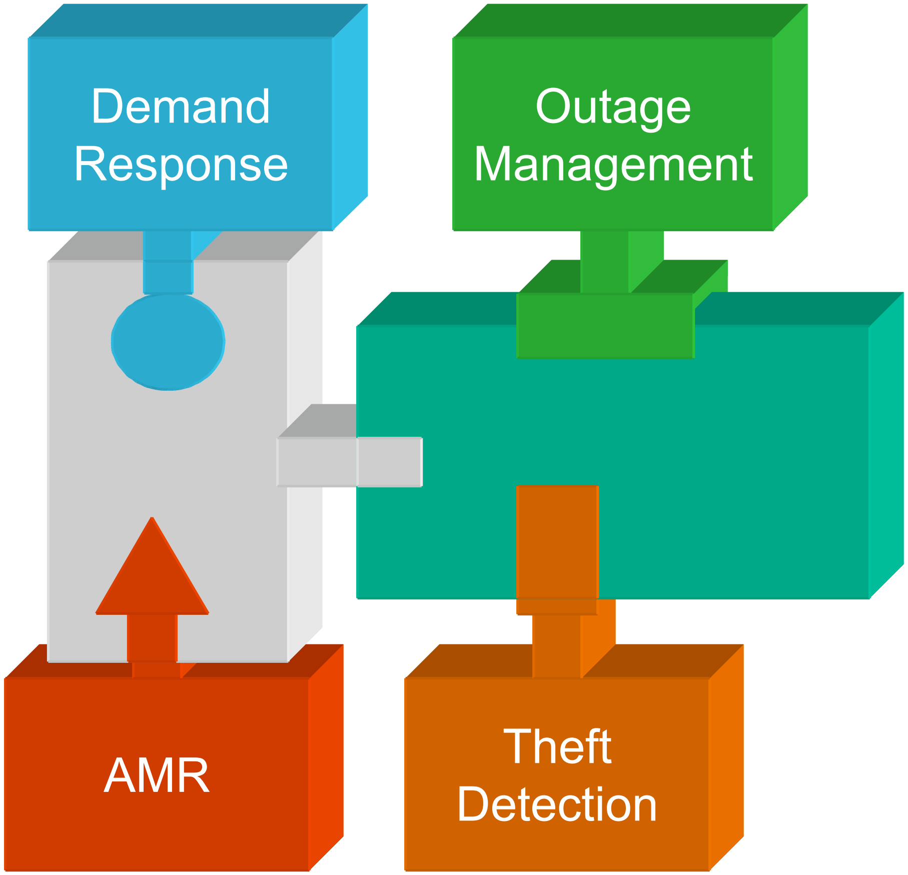
One-Off Integration



- Integration is typically done after the fact
- Cost is significant

HOW IT’S USUALLY DONE

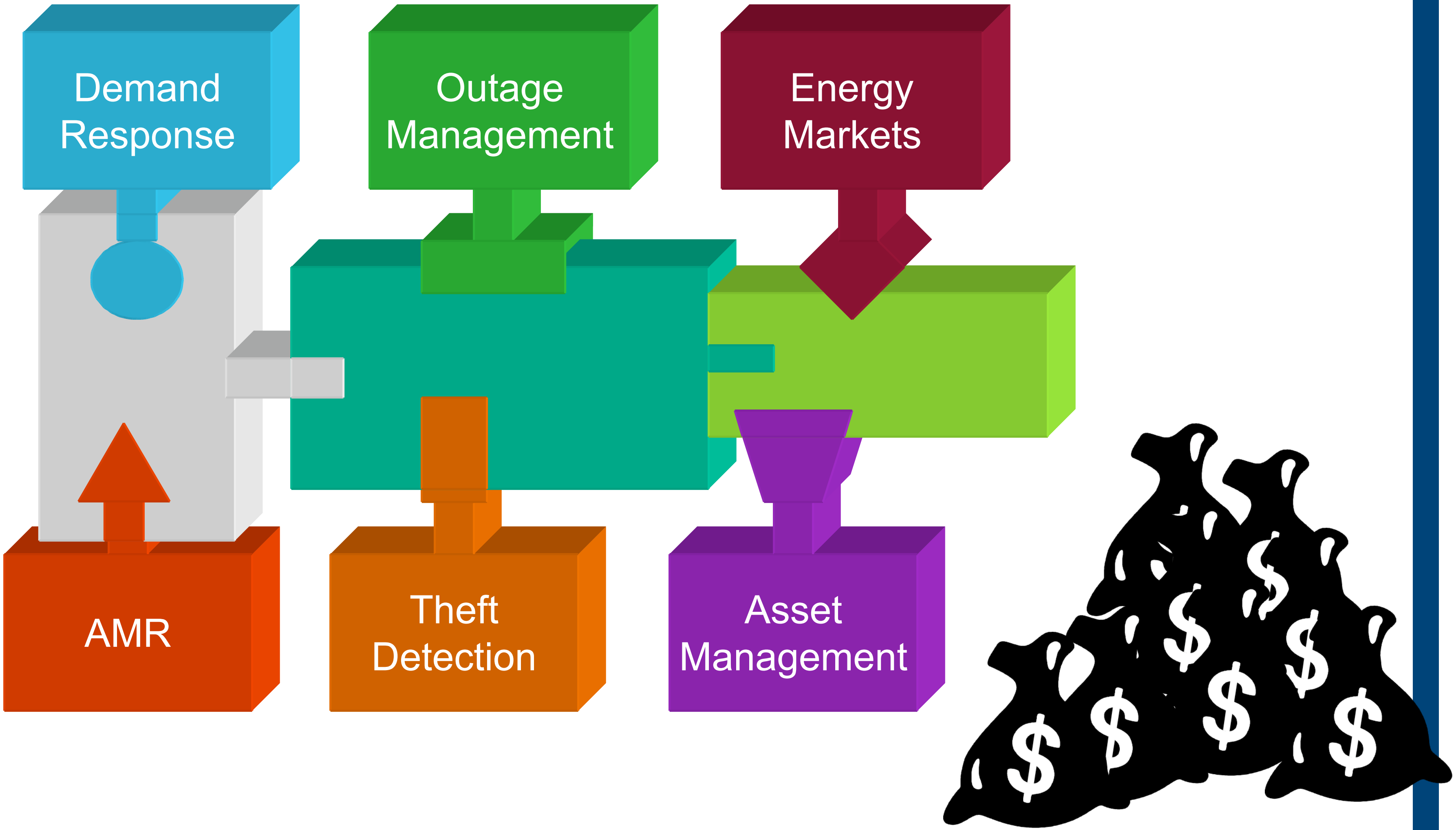
Doing it the Next Time



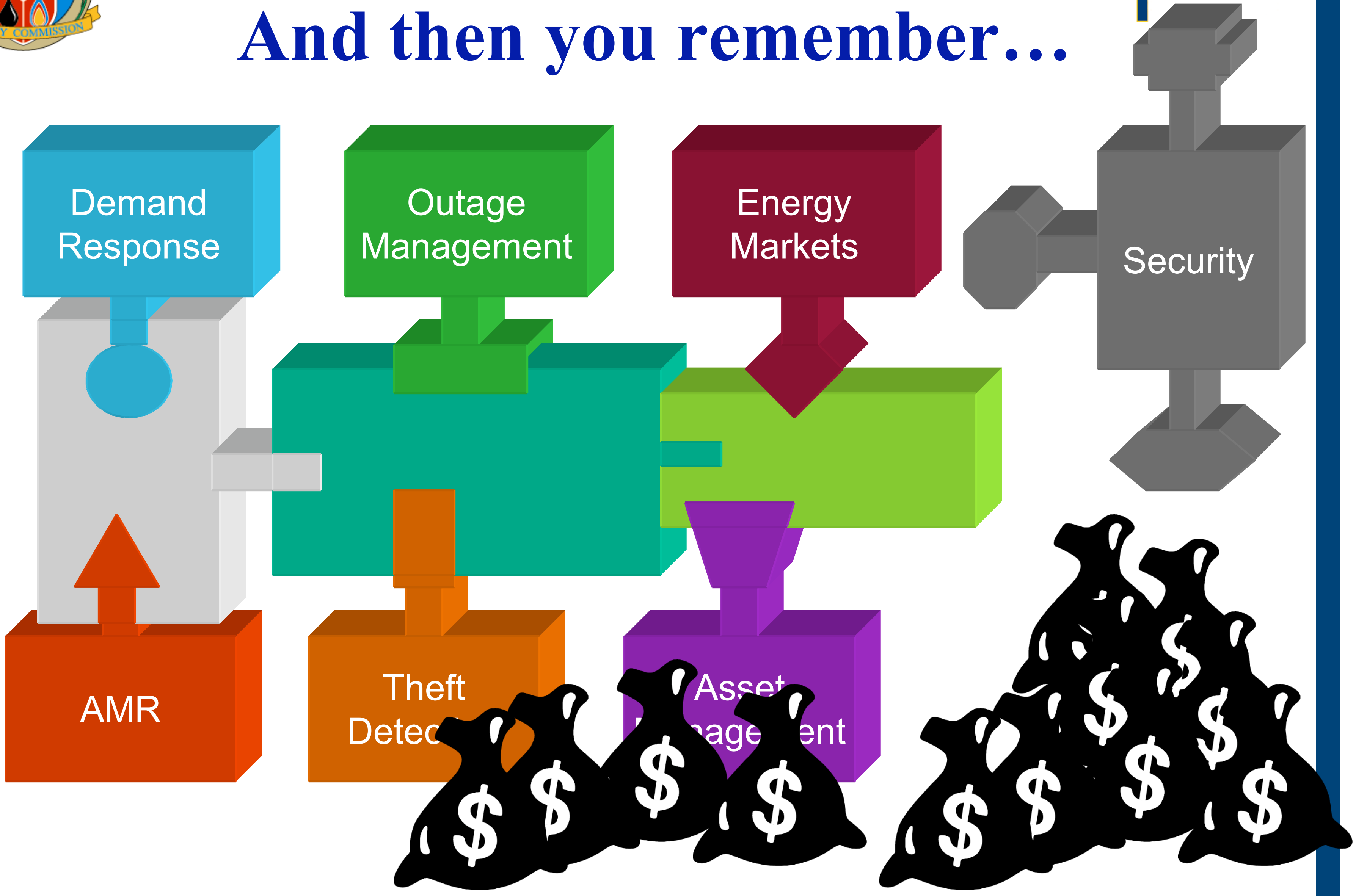
- Now want to link in new systems
- Must first make the old system expandable
- Then must do another “one-off” integration

HOW IT'S USUALLY DONE

And again…

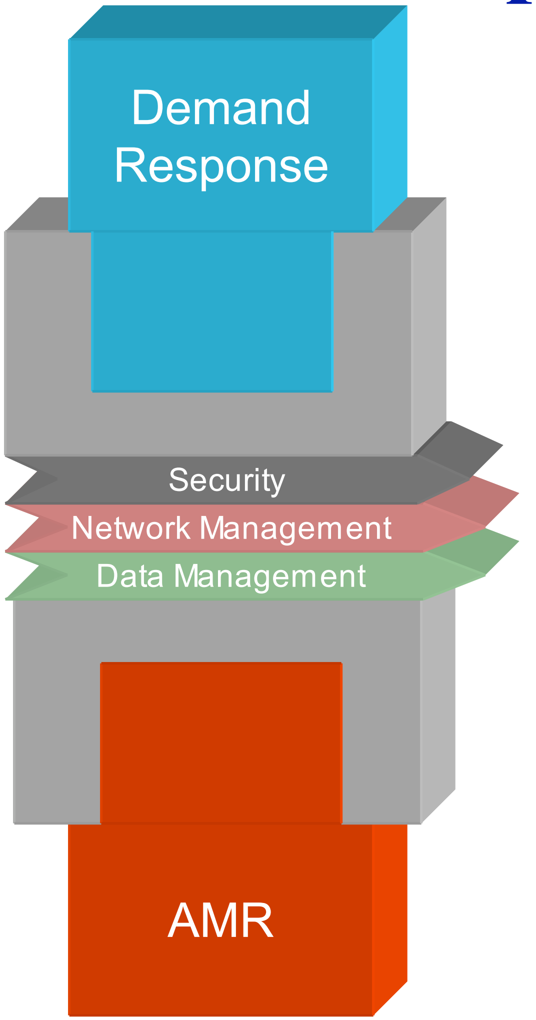


And then you remember…



DOING IT RIGHT:

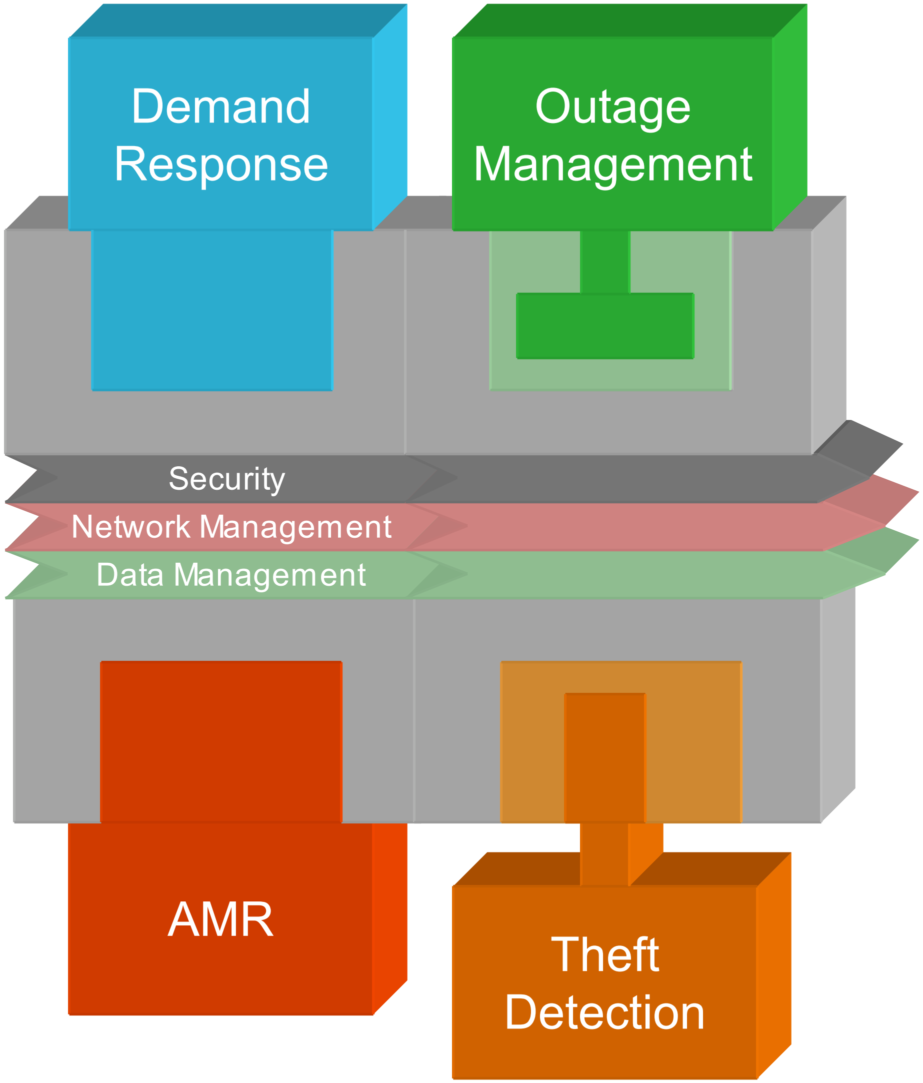
Top-Down Architecture



- Define standardized interfaces first
- Incorporate security, network management and other strategies right from the beginning
- Initial costs are a bit more than one-off integration, but not much more
- New applications can build directly to the new architecture

DOING IT RIGHT:

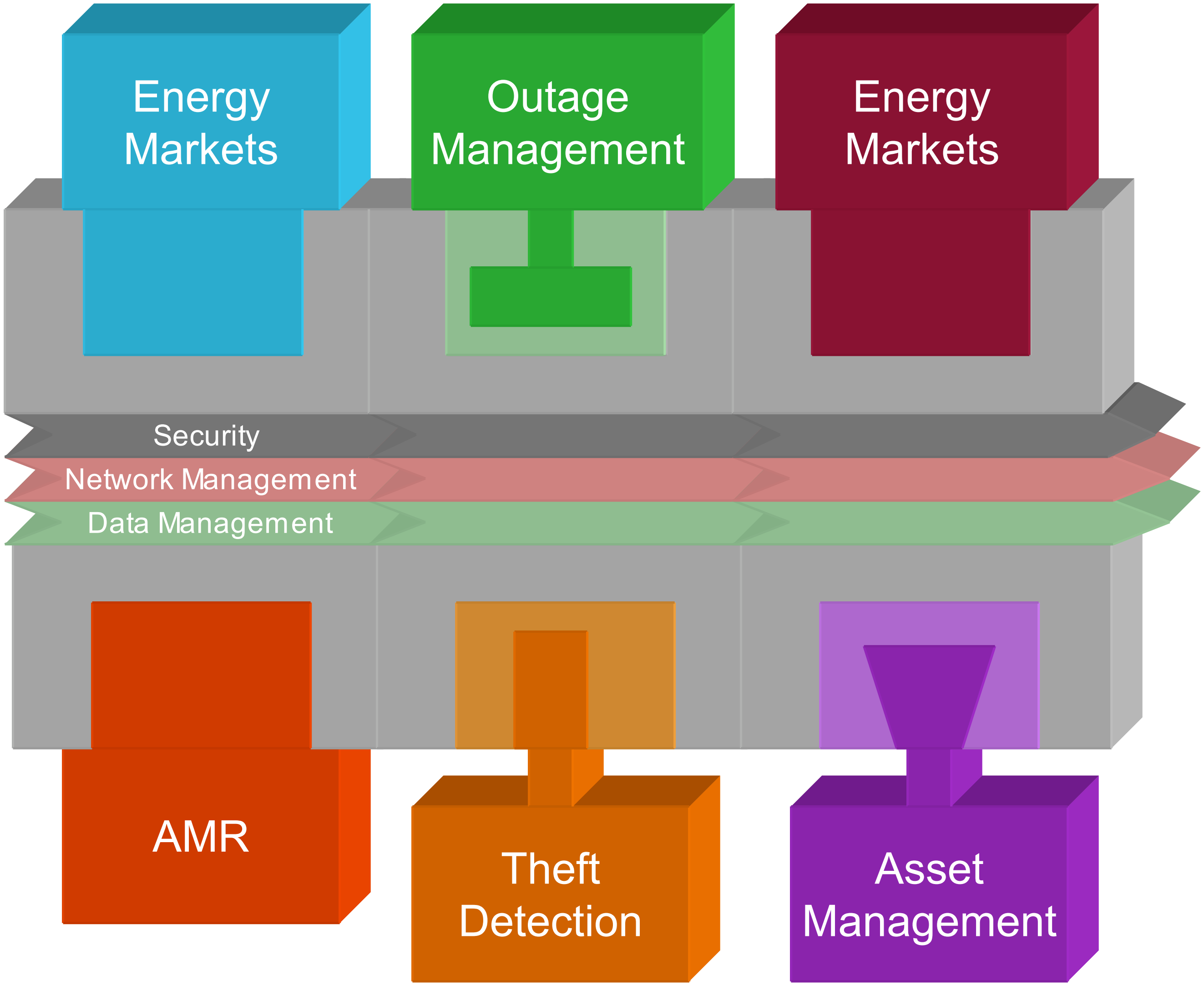
The Next Phase



- Can re-use the development from the first phase
- Expansion was expected
- Adaptation to legacy systems was planned in advance
- Overall costs much lower

DOING IT RIGHT:

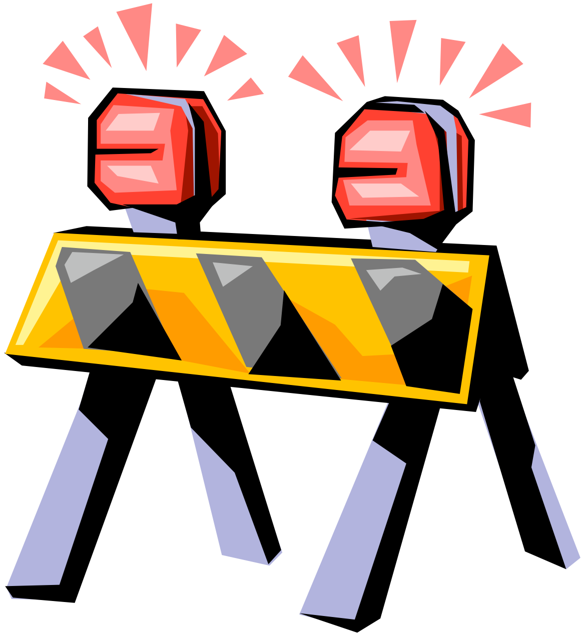
And so on…



- Benefits INCREASE with time
- Opposite of the old way

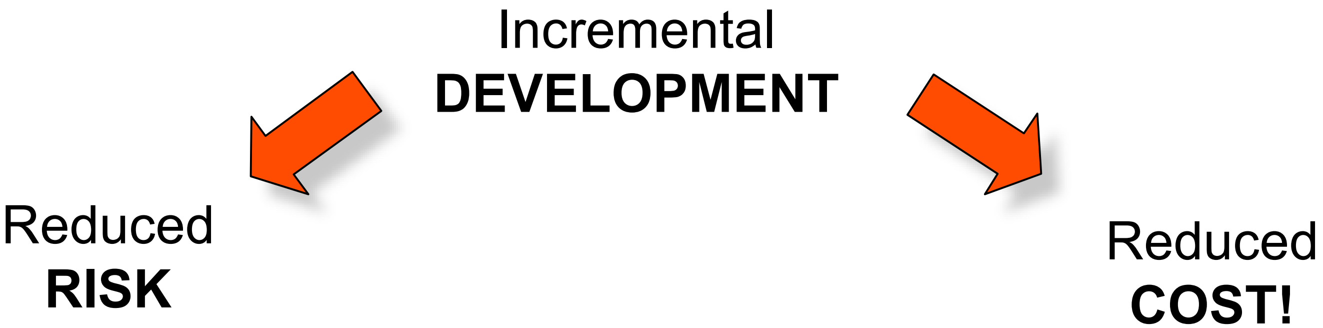
Barriers to Creating Integrated Systems

- *Organizational*
  - Corporate culture
  - Existing policies
  - Departmental “Silos”
  - Lack of human resources
- *Knowledge*
  - Missing applications
  - Missing algorithms
  - Lack of training
- *Systems*
  - Lack of standards
  - Too many “standards” to choose from
  - Missing products and services
  - Not applying formal methodology



The Value of Top-Down Integration

- Permits capital investment to be re-used
- Eliminates redundant effort and last-minute retrofits
- Prevents forklift upgrades
- Vital system-wide capabilities, like security, come standard
- Prepares the system for unforseen circumstances:
  - New technology
  - New applications
  - New organizational change



Incremental **DEVELOPMENT**

Reduced **RISK**

Reduced **COST!**