Software Needs and its Valuation in the Electric Power Industry

Marija Ilic, Carnegie-Mellon University

milic@ece.cmu.edu

Paper motivation

- Serious lack of user-friendly software for managing the complexity of real-life systems
- Design of software huge challenge, but the missed opportunities huge as well if not done
- Not necessarily a self-explanatory nor widely accepted statement

Some basic difficult questions

- How does it work today?
- What needs fixing?
- What are some possible performance metrics which provide us with quantifiable ways of showing improvements?
- Are methods under consideration capable of meeting the pre-decided upon performance criteria?
- What are systematic ways of deploying new technologies into the existing system without making the overall operations even more complex?
- How to integrate new in ways transparent and useful to those operating the system?
- How to provide policy and financial incentives for deploying the most effective technologies as measured in terms of pre-agreed upon metrics?

Paper outline

- **Something old: Is all well in today's operations and planning (emerging problems with reliability, missed missed opportunities for enhanced economic utilization; challenges to the existing software)**
- **Performance metrics for assessing value-added by the existing software**
- **Information and software specifications for reliable operations**
- **Something new: Operating and planning in the changing industry**
- **Needs for novel IT and software solutions**
- **Performance metric for assessing value-added by IT and software for the changing industry**
- **Information and software specifications for the changing industry**
- **Technical and economic policy roadblocks to software deployment in the changing industry**

Something old: Is all well in today's operations and planning

- Operators and planners have two basic objectives, namely to serve customers reliably and at acceptable cost.
- Operations and planning inter-dependent (planning assuming operating practices; and, vice versa, operating practices assuming planning principles)
- In the past, operations relatively straightforward based on robust design which enables many simplifying assumptions in operations (localized response to system failures; semi-stationary feedforward for given demand forecast; hierarchical temporal and spatial separation)
- Even during equipment failures sufficient reserves and pre-planned procedures almost always sufficient for acceptable service.
- Economic utilization achieved using very simple ED computations for real power; sufficient support for reactive power through design.
- Utilities run by the human experts w/o critical reliance on on-line software use and extensive automation.

Simplifying assumptions no longer justifiable

- The interactions among utilities and within utilities themselves have become more complex than in the past, and are beyond human's ability to manage;
- Economic factors no longer allow robust operations through design; need for just-in-time (JIT) and just-in-place (JIP) decision-making.
- JIT and JIP services require much on-line sensing, monitoring and software-based decision making
- Unexpected network system response as utilities trade for economic reasons (patterns very different than what was pre-agreed on; load decrease could cause continuing decrease in frequency and voltage, contrary to the operator's intuition; wide-spread backbone effects of equipment failures leading to cascading failures)

Fundamental need for on-line information processing

- Implementation of pre-agreed performance metrics over a broad range of conditions (short-term enhanced reliability; enhanced short-term utilization of existing resources; enhanced long-term service to customers)

- The overall problem of operating the system away from “nominal” conditions

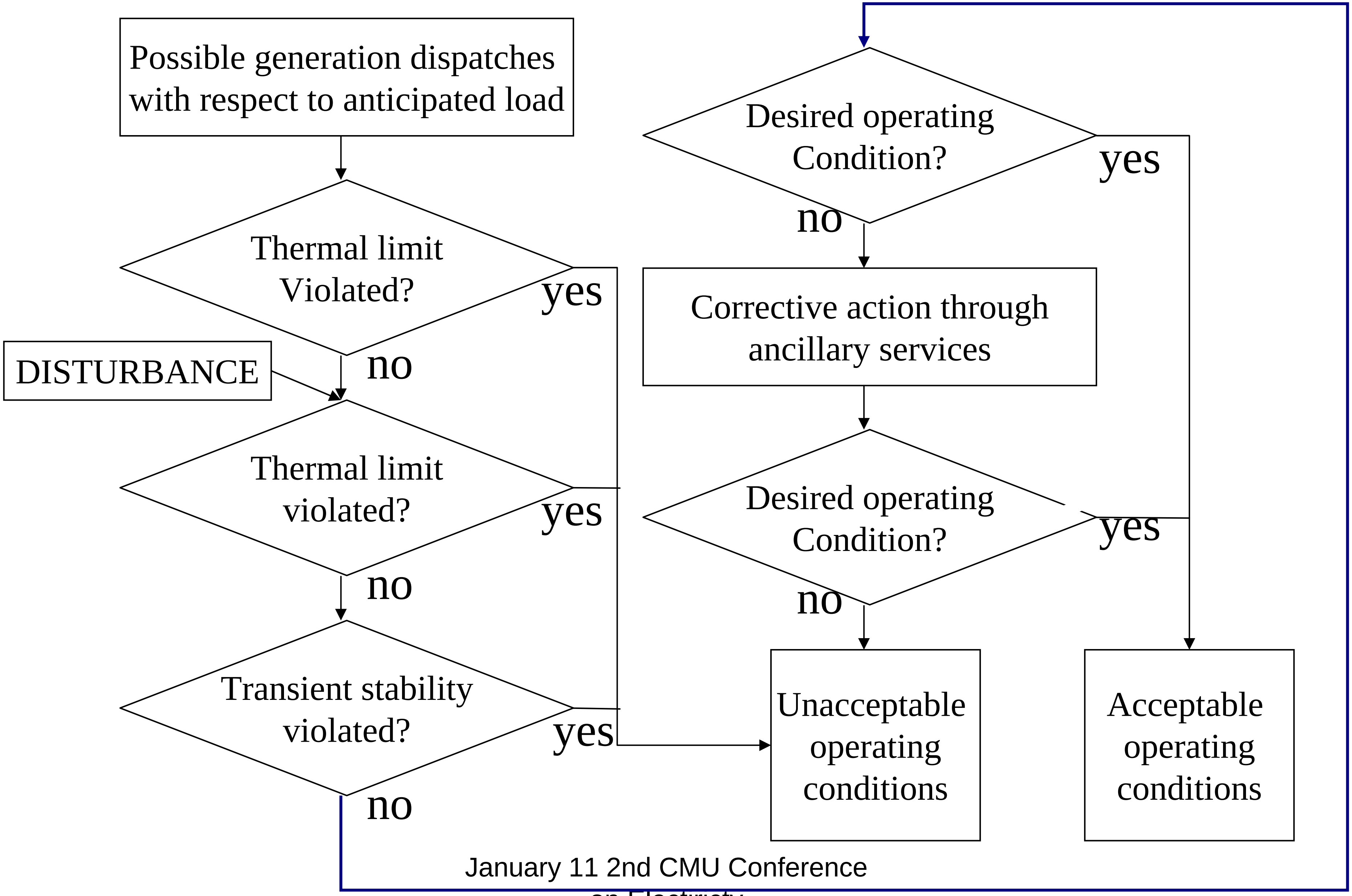
- These are not directly interpretable in terms of (N-x) reliability standards (particular amount of reserve does not necessarily guarantee pre-specified LOLE, and, more generally QoS.

Software specifications for facilitating reliable operations

- Relying on software risky unless one has robust and easy-to-use software
- Today's software does not meet these requirements
- Software has evolved by solving particular sub-problems under strong (often hardware-ensured assumptions)
- In order to implement (N-x) reliability standards one needs a dynamic shell (architecture) for integrating the existing modules with well-defined performance criteria and internal logic for relating various software modules (hard, and loaded with open problems)

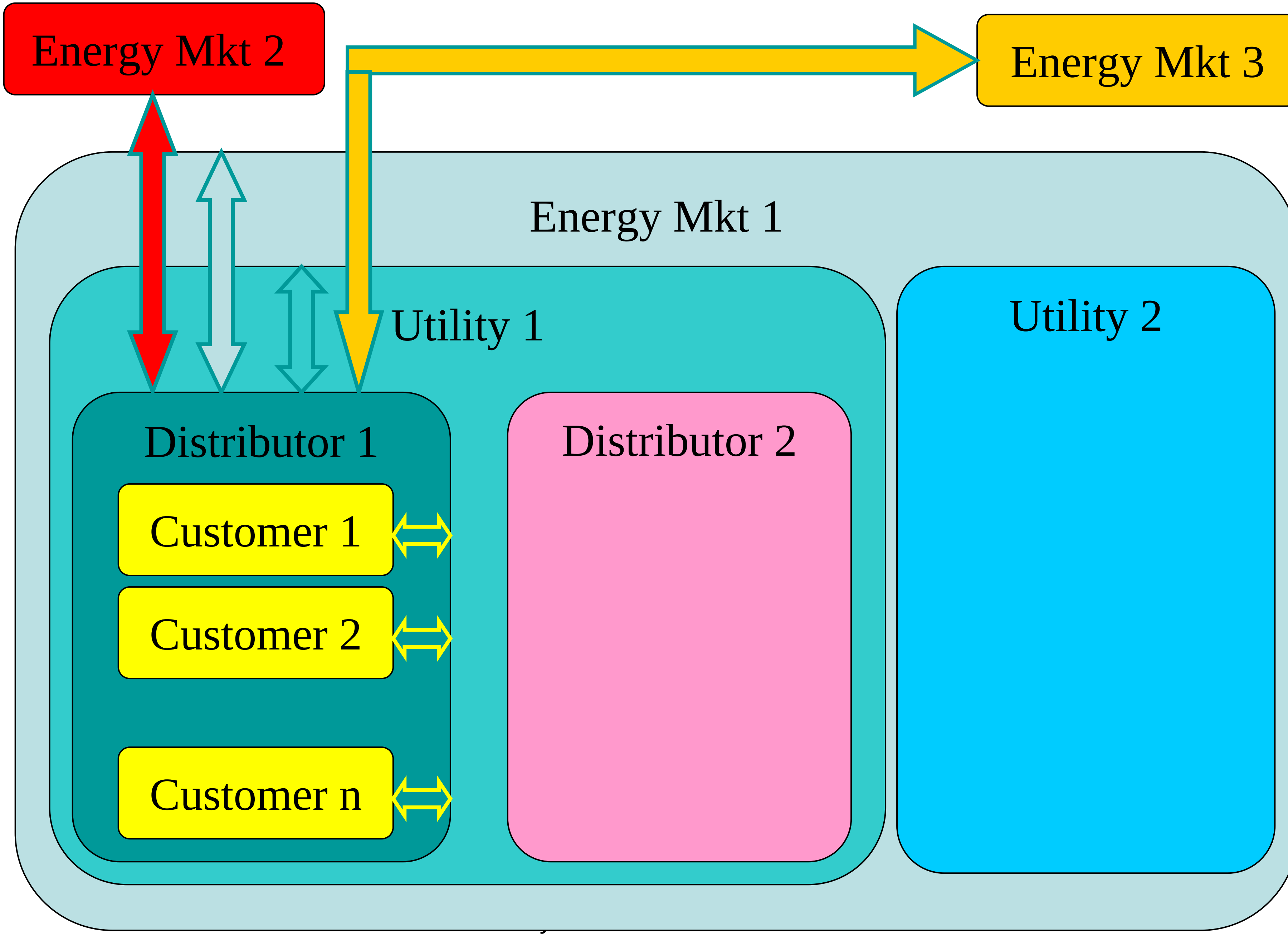
An illustration of performance metrics

- At the shell (architecture) level: Customers served according to QoS (TOU service, probability of not being served)
- Designing a sufficiently general architecture for minimal coordination (logic) among the internal software modules is one of most difficult tasks
- The second hardest task is processing of huge amount of data into used and usable set of recommendations to operators for ensuring QoS as conditions vary
- A well-functioning shell should be sufficiently flexible to allow for many solutions (technologies, hardware and software) which, jointly, result in comparable performance at the shell level


Possible generation dispatches with respect to anticipated load
Thermal limit Violated?
yes
no
DISTURBANCE
Thermal limit violated?
yes
no
Transient stability violated?
yes
no
Desired operating Condition?
yes
no
Corrective action through ancillary services
Desired operating Condition?
yes
no
Unacceptable operating conditions
Acceptable operating conditions

Something new: Operating and planning in the changing industry

- Distributed performance metrics, associated with the candidate new technologies and/or unbundled entities
- IT and software capable of accommodating these distributed performance metrics, and extracting their value-added to the performance metrics at (various) layers of the shell
- Definitive need for extracting value-added through distributed JIT an JIP performance within the given contextual, spatial and temporal interplay
-



on Electiricty

Software specifications for the changing industry

- Need for flexible protocols (easy-to-reconfigure) to provide bundled services (energy, delivery, Quality of Service (“QoS”))
- Multi-layered protocols are essential to create software and hardware development incentives, providing compelling value proposition to customers

- We are working toward protocols and software for dynamic (electric) energy control, allowing “true” customer choice and enhanced, sustainable business models for distributors, utilities and markets

The underlying change of paradigm [1]

- The electric power industry processes are a result of numerous small decisions/actions; sharp contrast to the old industry
- Micro-level actions contribute to significant change at the macro-level
- Economies of scope gradually replacing economies of scale
- New opportunities are based on this change; however, current operating/planning/design practices do not support this change

[1] Jelinek, M., Ilic, M., ``Strategic Framework for Electric Technologies:Technology and Institutional Factors and IT in a Deregulated Industry'', NSF Workshop, 2000.

A Dynamic Energy Control Protocol to support the new paradigm [2]

- Defines relations between physical, information and financial processes, across entire industry;Allows for flexible, creative decision making within these well recognized relations; Software based, with various degrees of automation; could accommodate many users
- Without these, the customer choice is not sustainable – "market is not ready"; many business consequences
- 2] Ilic, M and J.., Dynamic Energy Control Protocols for the Changing Electric Power Industry, Power Systems Computations Conference (PSCC), Barcelona, Spain, July 2002 (submitted)

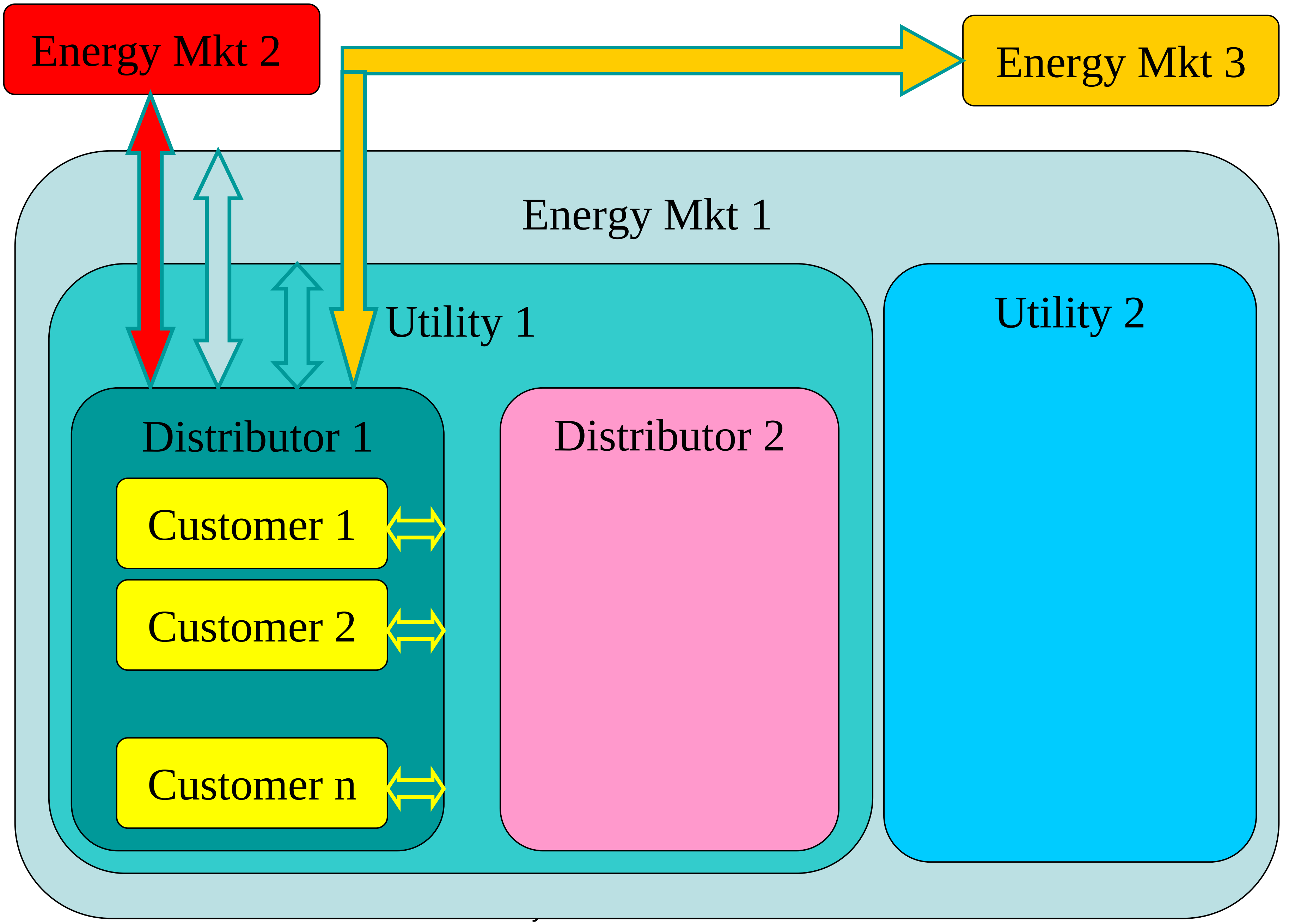
Review of the major broken links in the changing industry

- No well-defined relations/values across boundaries of industry organization (customer to distributor, distributor to utility, utility to wholesale market, market to market)
- Broken link between wholesale energy market and customers (many examples: (a) level of reserve and customer load profile and willingness to be interrupted; (b) charges for delivery T/D; c) value of backup power; d) tradeoff between DG and wire value to the customer)
- No relations between operations and investment processes
- No well-defined choices to the customers (in terms of financial and physical arrangements)

The Role of Multi-Layered Dynamic Energy Control Protocols (DECPs) and Software

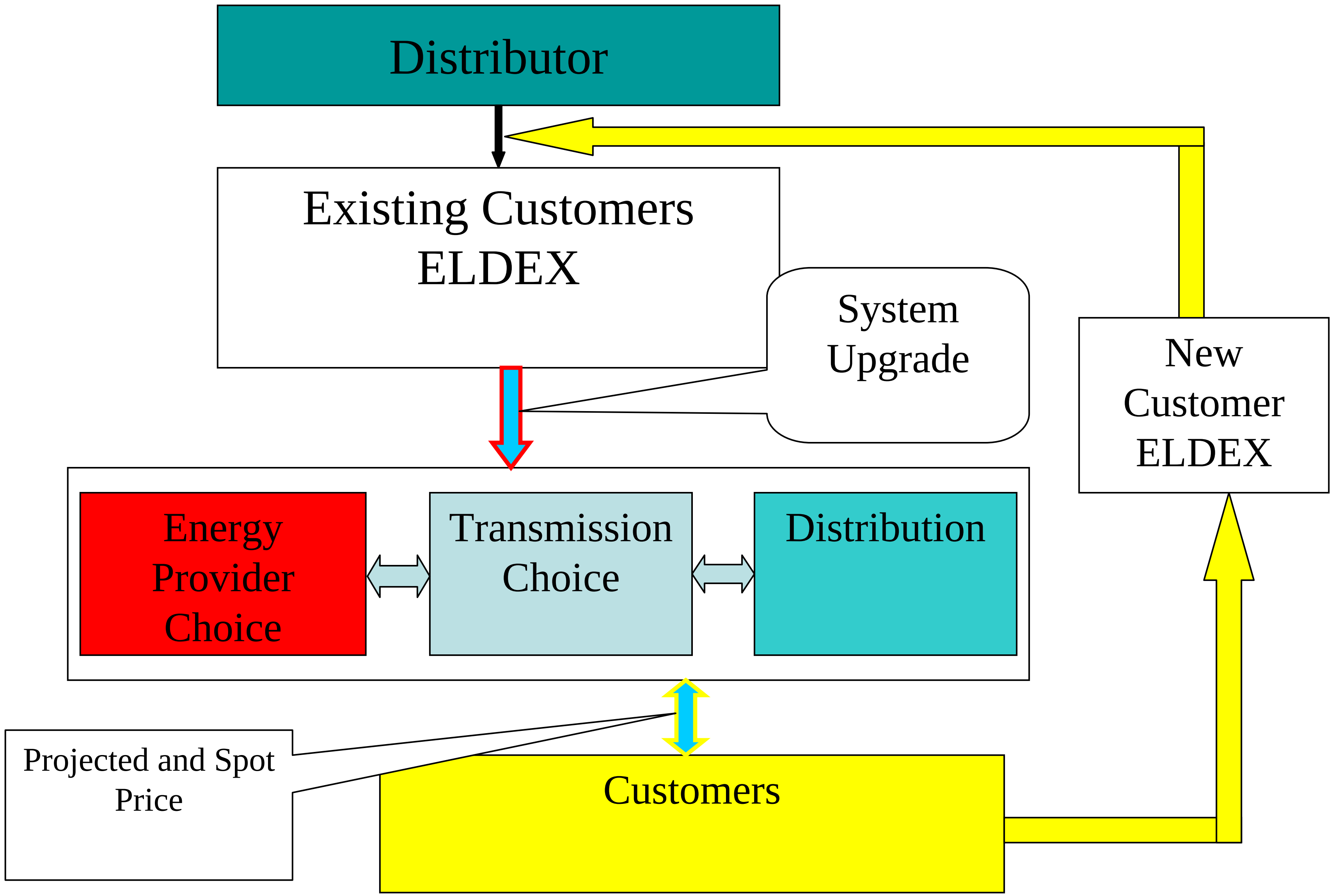
- To be used by the customers, as well as by the various providers of services to the customers in identifying right incentives

- Focus on proactive distributors for facilitating true customer choice

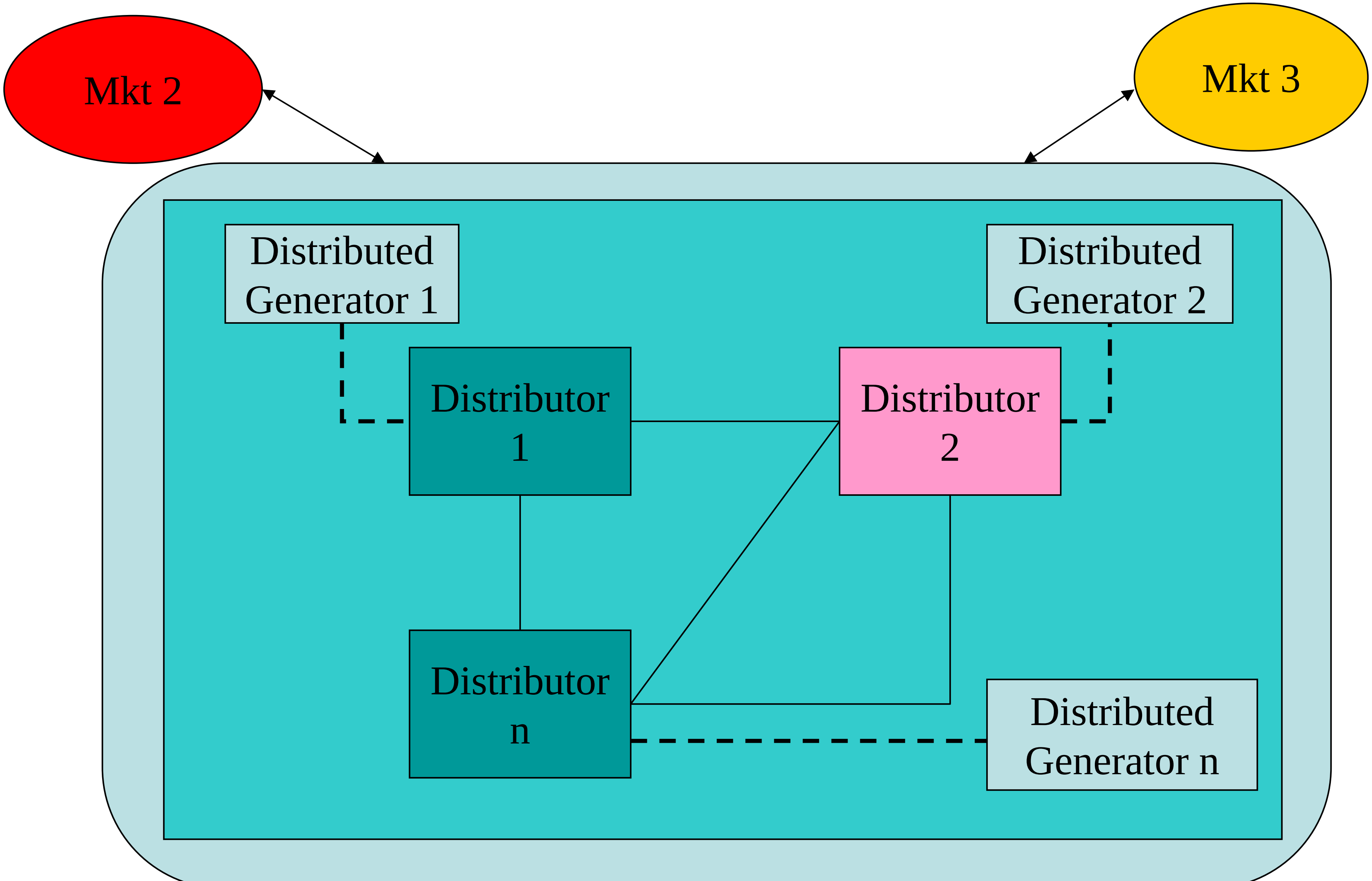


on Electiricty

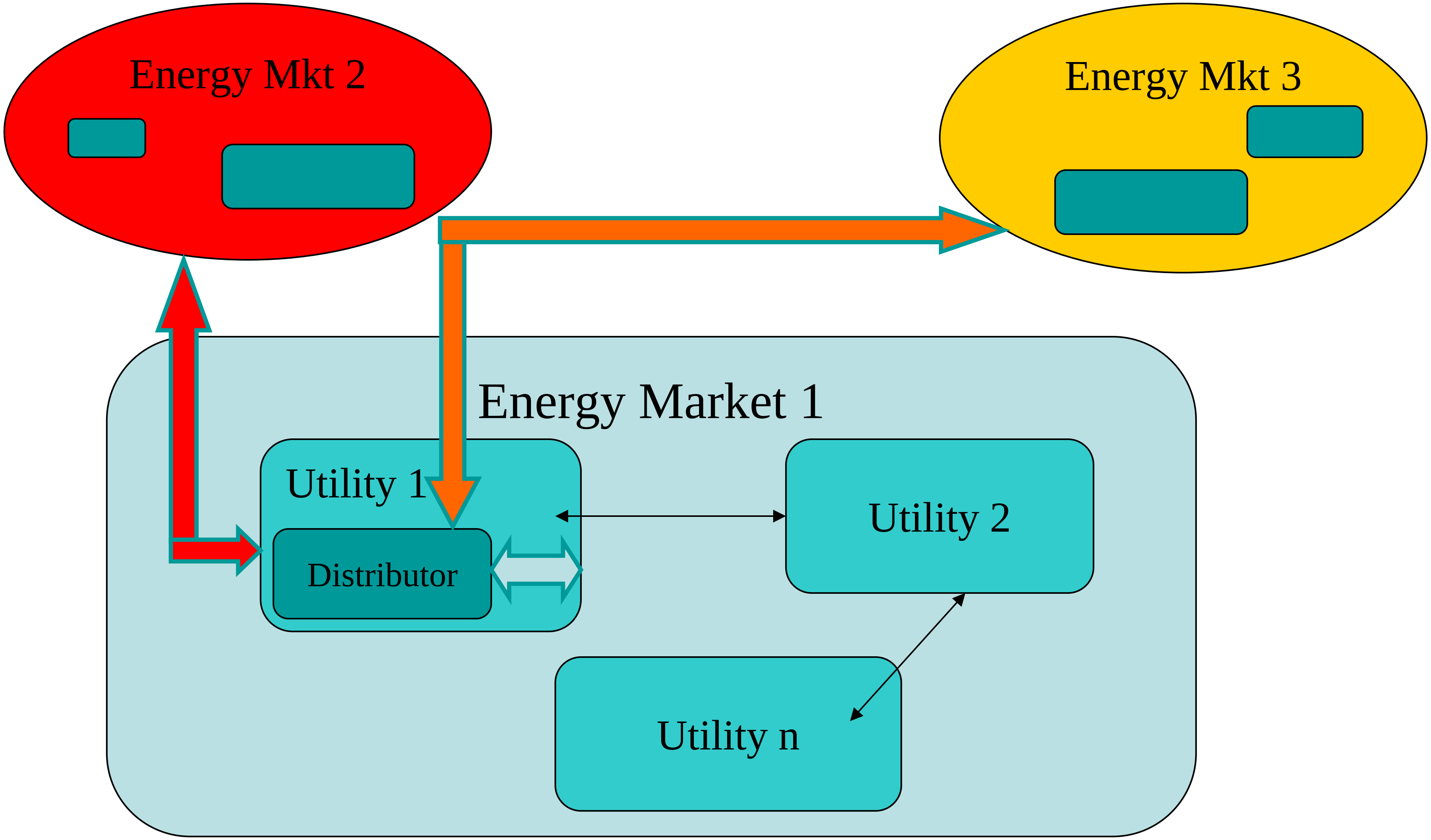
Dynamic Protocol --- Distributor Level



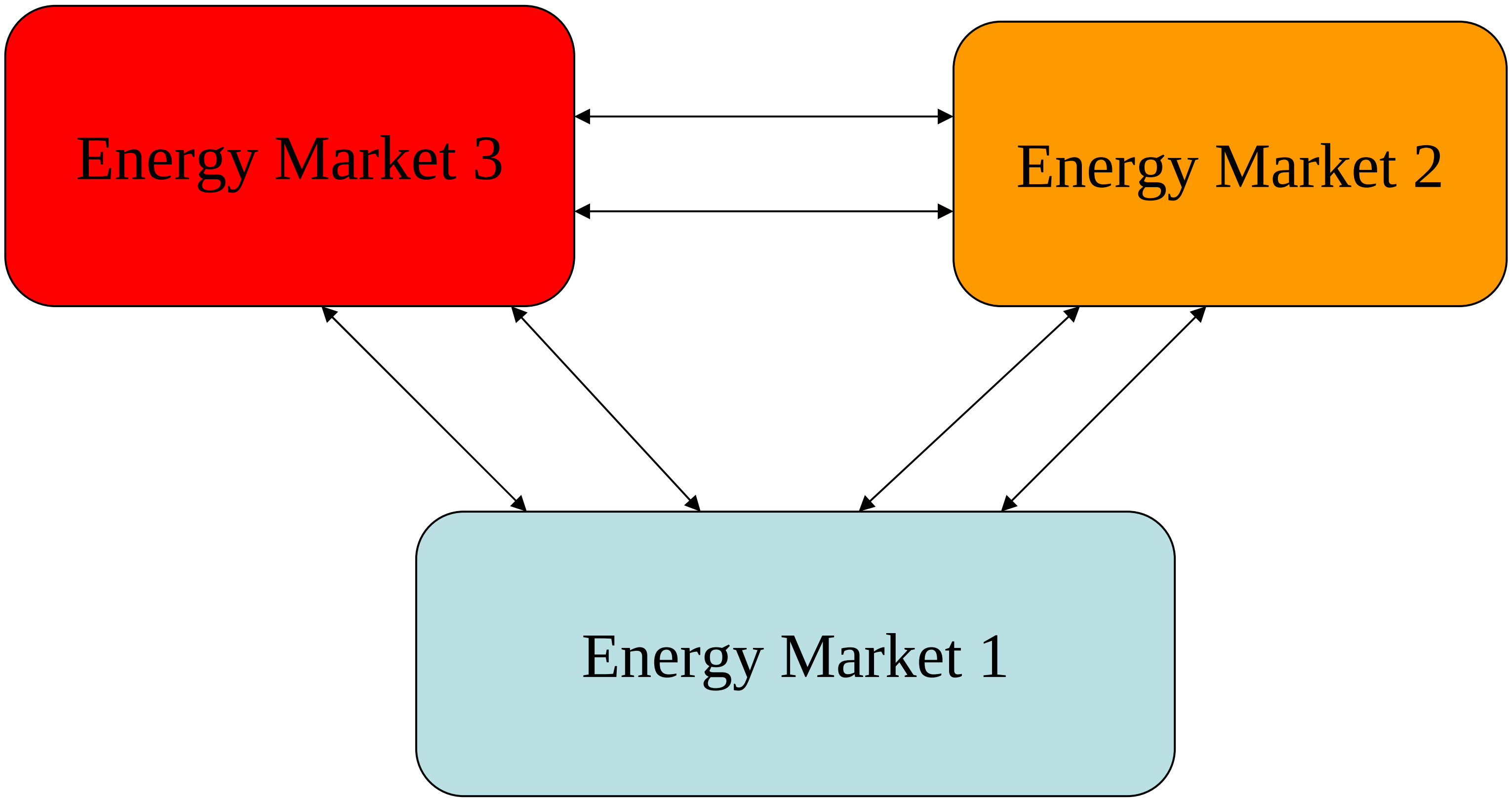
Dynamic Protocol --- Utility Level



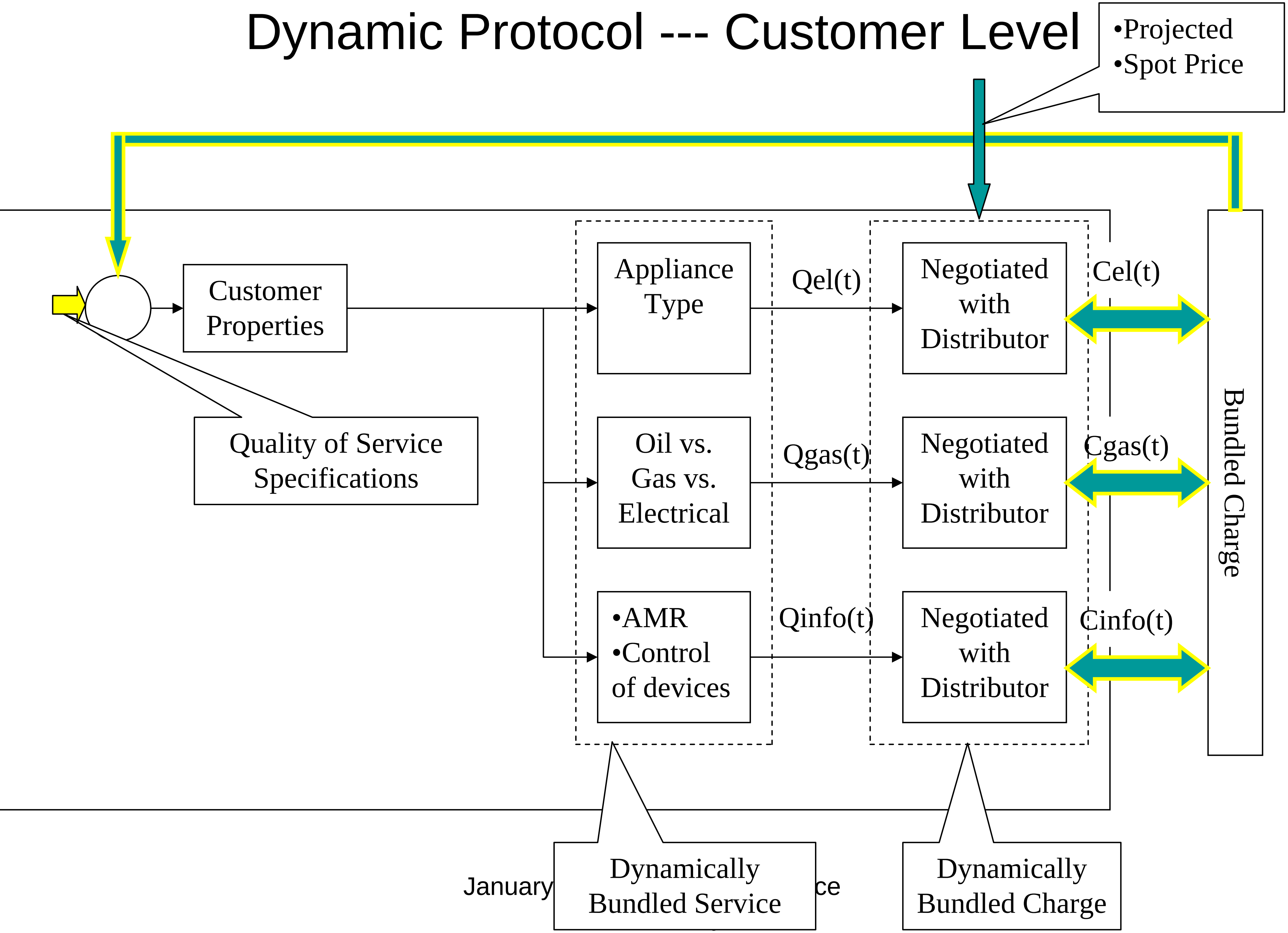
Dynamic Protocol --- Energy Market Level



Dynamic Protocol --- Multi Market Level



Dynamic Protocol --- Customer Level



Need for software development – Distributor level

- For supporting Energy Demand Experiment (ELDEX), to determine how much users value different QoS; for implementing differentiated QoS (AMRs,various controllers and platforms).
- For adaptive aggregation of customers, and dynamic accommodation of the new ones over time
- For adaptive contractual arrangements in face of various uncertainties (fuel, spot price, forward price, delivery charges,environmental costs)-with the customers and the higher level players (utility, energy market,multi-market) ; ranging from simple, insurance-like, through speculative with well understood risks)
- For valuing contracts as a function of technology, and vice versa; optimum investment time decisions.
- Dynamic integration of the above modules.

Need for Software development—Utility level

- For managing the system reliably, with distributors actively changing T/D needs on behalf of their customers, and with other utilities/markets wheeling over their area
- For implementing active information protocols with the distributors, on one hand, and higher levels, on the other (markets)
- For optimal investment decision making (timing, type—DG vs wire)
- For having flexible value-based T/D tariffs (spot, and longer-term,capacity,reliability) capable of supporting feasible investments
- Dynamic integration of the above modules.

Need for software development—Market level

- For managing the system reliably, by dynamically interacting with the utilities, distributors, and the neighboring markets (changing rigid technical standards, which do not allow for load and T/D demand control through pricing).
- For meaningful tariffs which relate value of the system to the end-users and the investments made (including unbundled valuation to the system users and the T/D entities)—explicit capacity,reliability valuation
- For dynamic integration of these modules.

Need for software development—Multi Market Level

- Software for facilitating inter-regional deals reliably

- Software for valuing wheeling support (wires, DG and other technologies) to the inter-regional transactions

- Software for interacting with distributors, markets, utilities

- Dynamic integration of these modules.

Summary

- There exist a variety of ways for designing software architecture capable of meeting requirements set by the industry practices, regulatory regimes and the evolving technologies.
- Need for robust software for catalyzing changes as specified by the regulatory and other rules.
- A dynamic multi-layered IT architecture (shell) needed under both regulated and changing industry paradigms (to integrate disruptive technologies, and/or to managed seemingly unbundled industry) for implementing well-defined performance