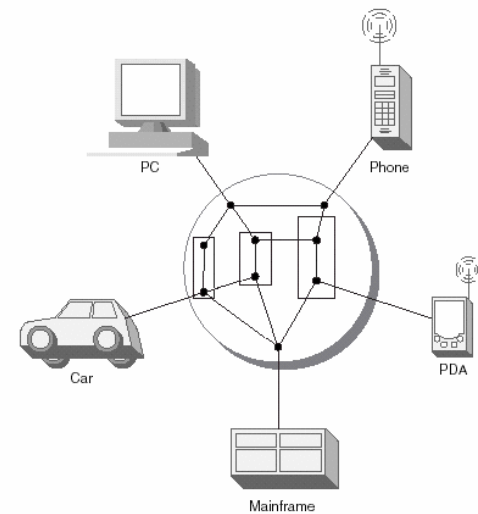


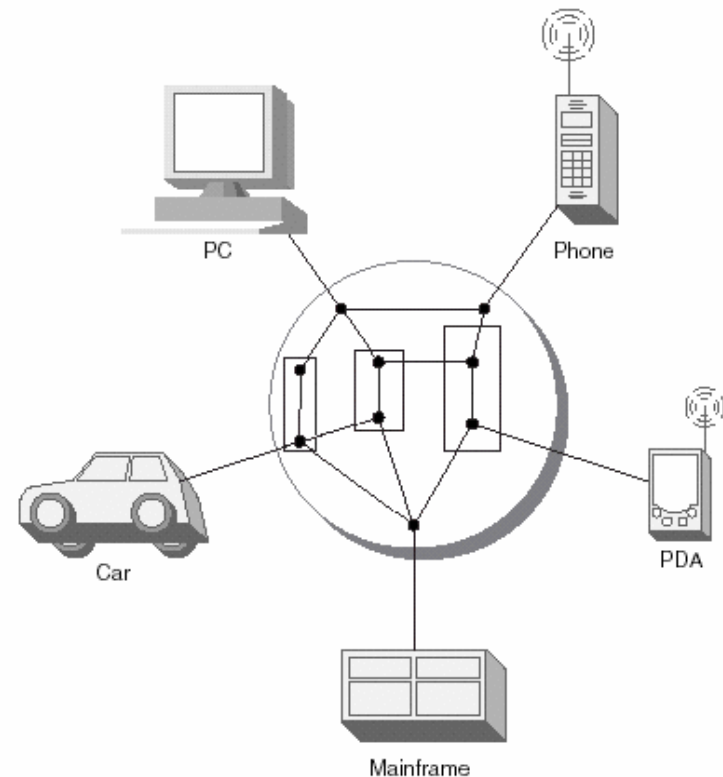
Feature Interactions in Web Services

Michael Weiss
Carleton University



Web Services

- Web services are a way of *packaging* and *publishing* functionality to the network for use by other applications
- Web services can aggregate other web services from a *web of services* to provide a higher-level set of features



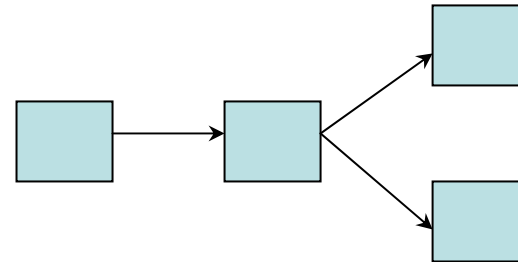
Web Services

- Paradigm for constructing distributed applications
 - Lack of centralized control (# businesses)
 - Diversity of technology platforms
 - Rapid evolution of business environment
- Much work to date on low level concerns
 - Publishing, discovering, and invoking of services
- Service integration raise significant challenges
 - Little research on managing interactions

Composition of Web Services

- Explicit composition

- Flow

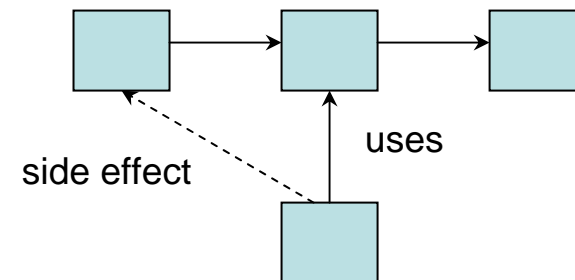


- Implicit composition

- Parallelism



- Side effect



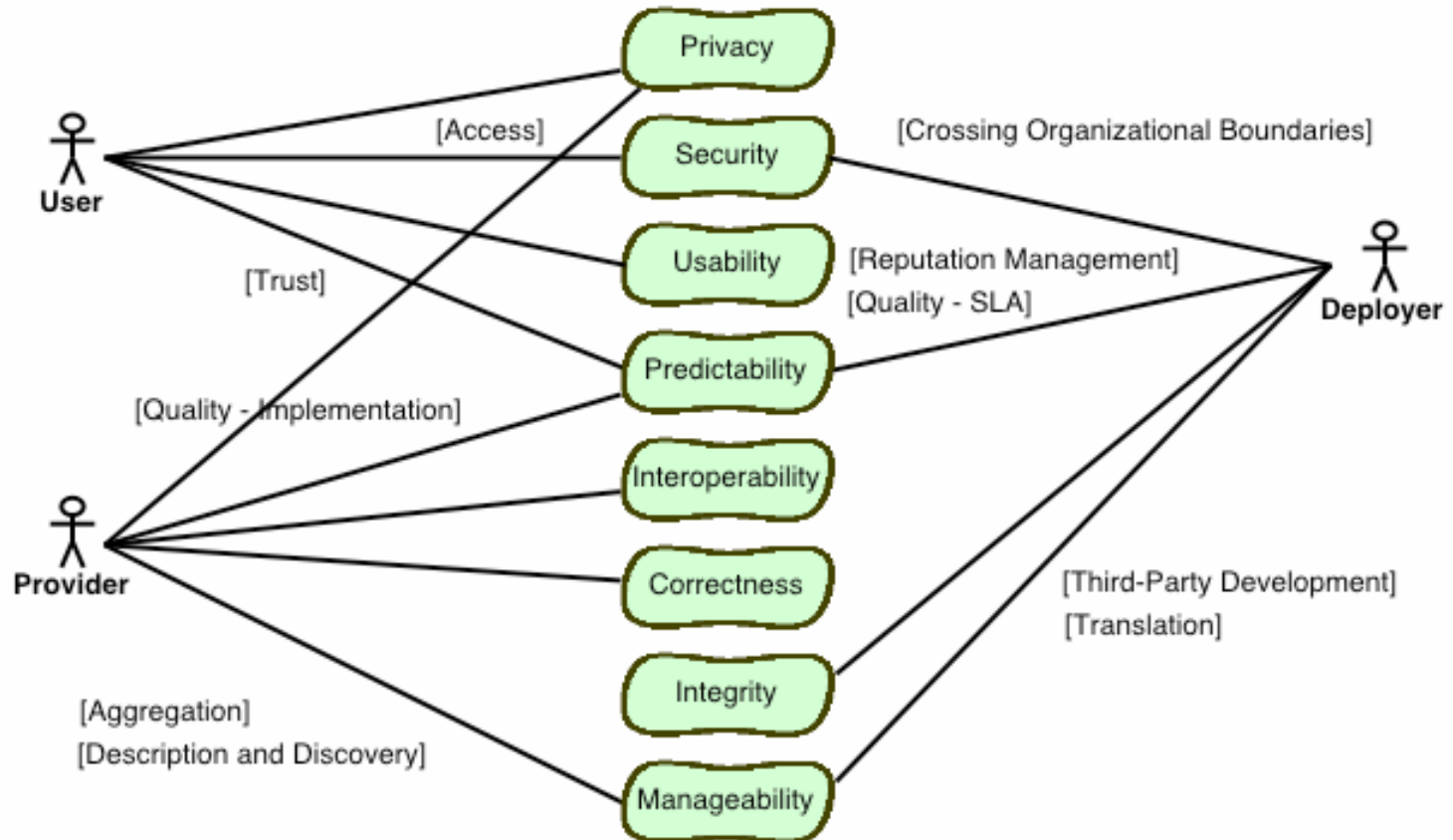
Feature Interaction Problem

- Coordination of features so that their composition achieves the desired result at application level
 - We use "feature" and "service" interchangeably here
- Root causes (from telephony)
 - Conflicting goals
 - Competition for resources
 - Changing assumptions
 - Design evolution
 - ... ? (web services)

Functional and Non-Functional

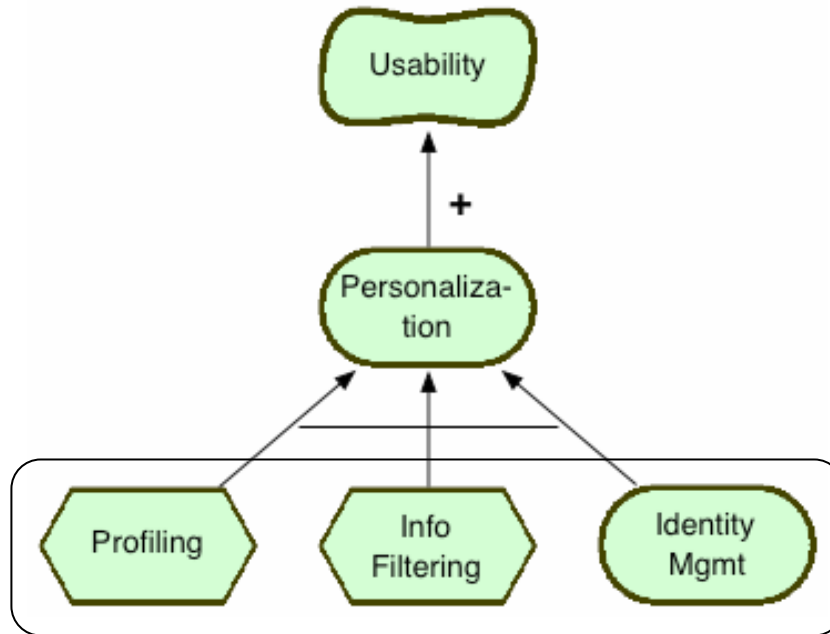
- Most interactions studied of functional nature
 - eg between (the functions of) CW and CFB
- However, each functional composition impacts the satisfaction of non-functional requirements
 - Mostly neglected by current research!
- It makes sense to talk of *non-functional* features
 - Usability, privacy, availability, security, ...
- Feature interactions in web services not really studied (some work on SoC, AOP, ...)

Non-Functional Features



Example: Personalized Services

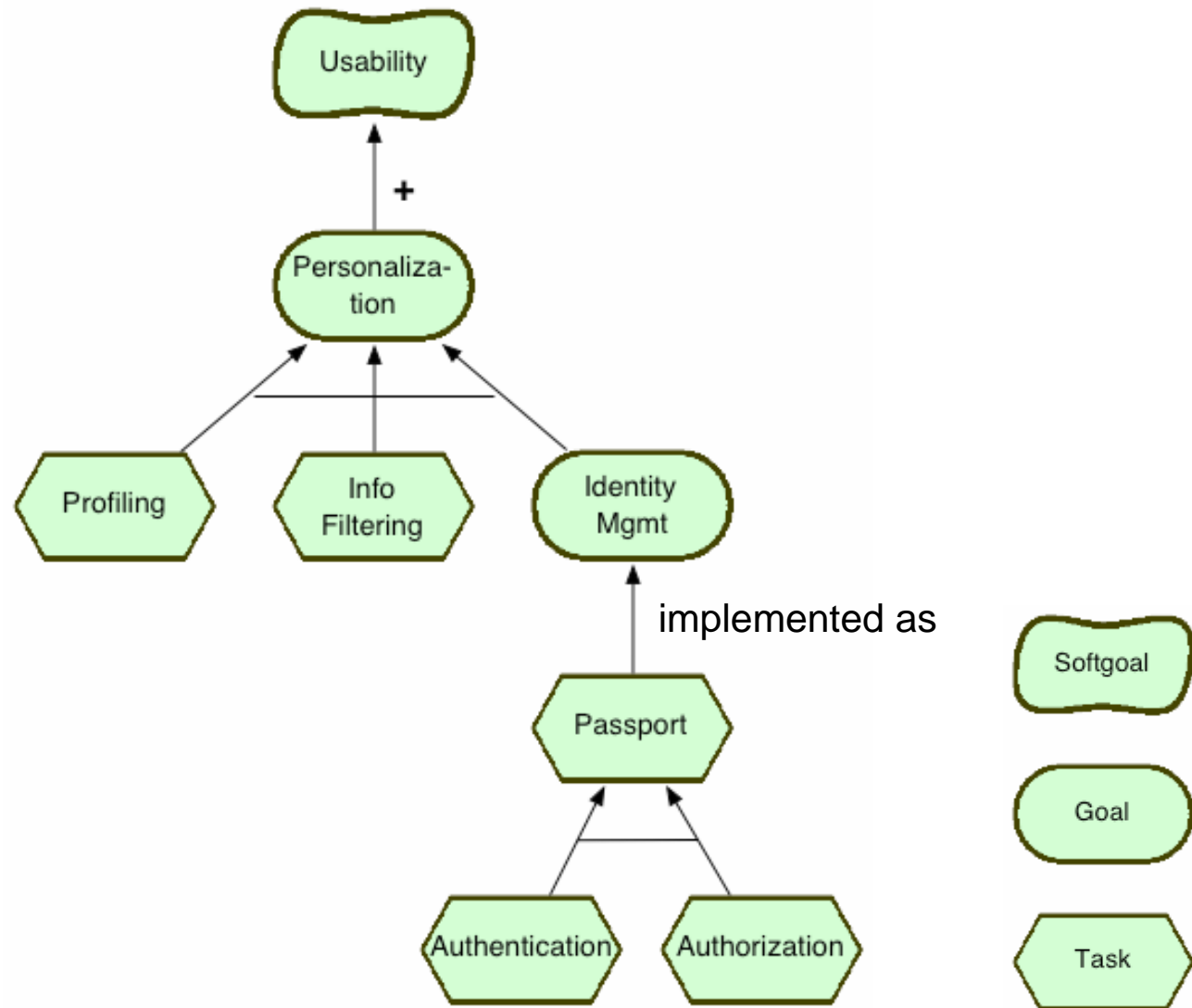
- m-Commerce
 - Personalization through information filtering
 - Based on user identity, profile, and the user's location
- Many design issues
 - Dynamic assembly
 - Trust (service quality, bias, ...)
 - Privacy concerns
 - Security
 - Semantic ambiguity
 - ...

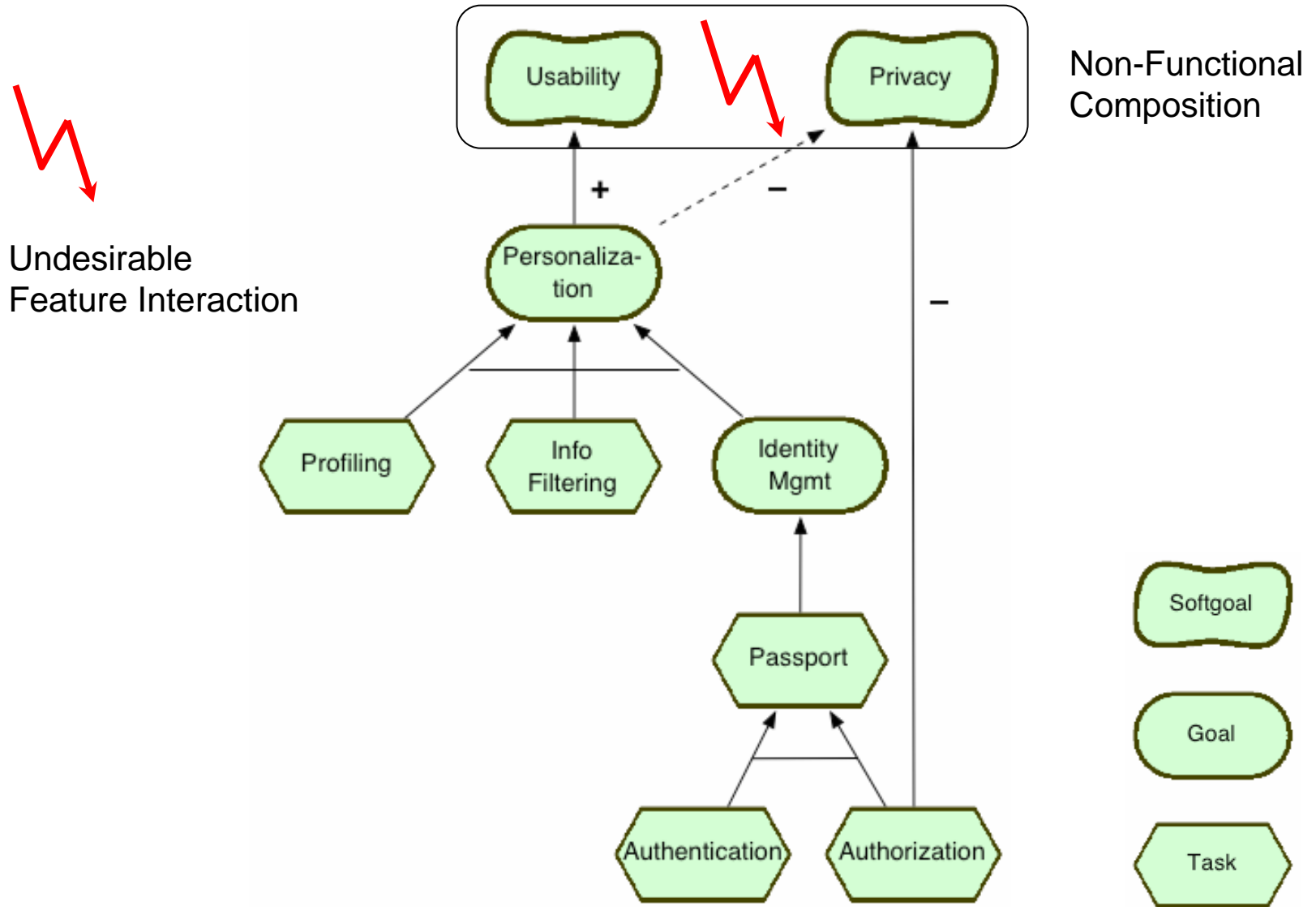


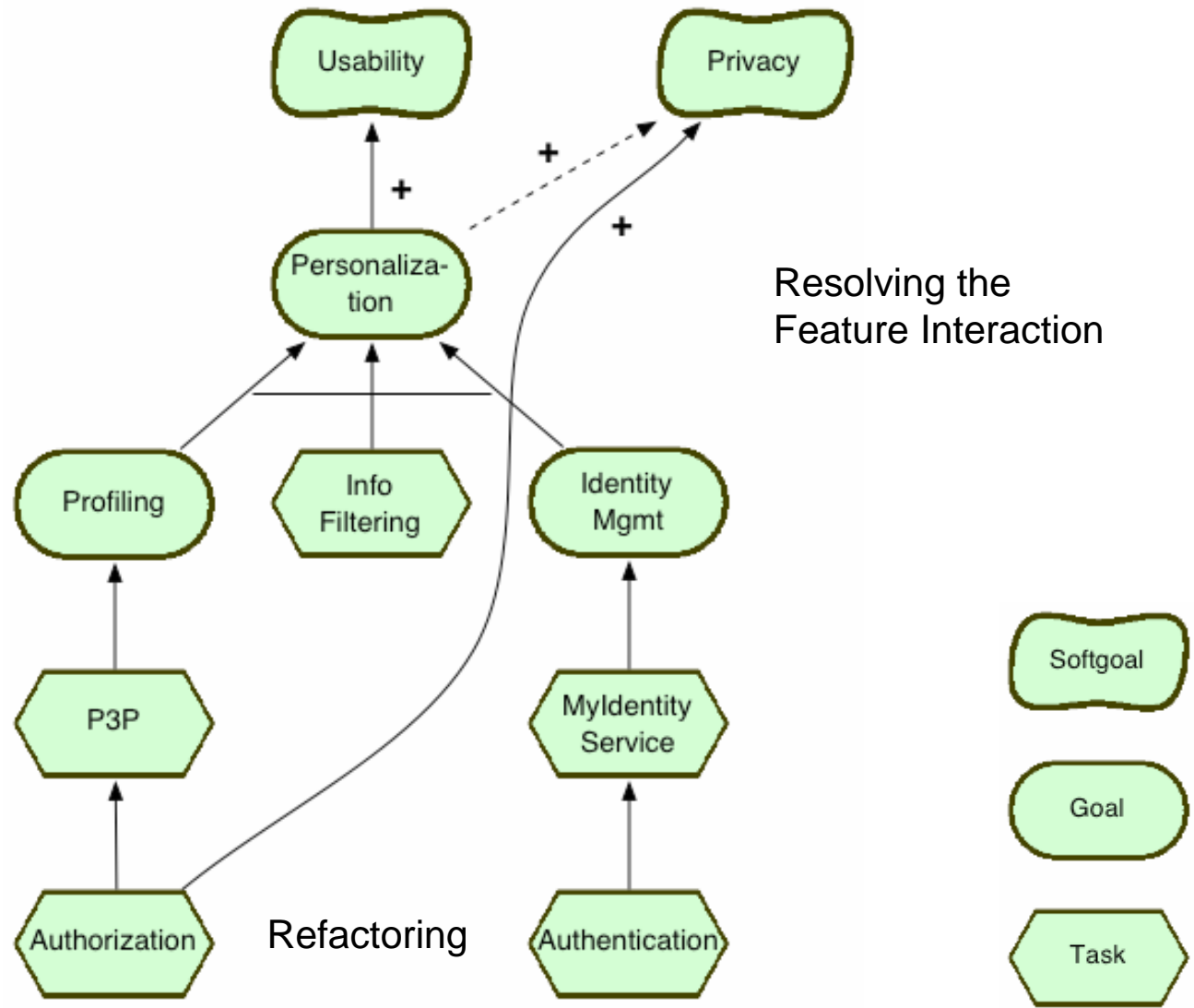
Functional
Composition

Notation: Goal-
Oriented Requirements
Language (GRL)










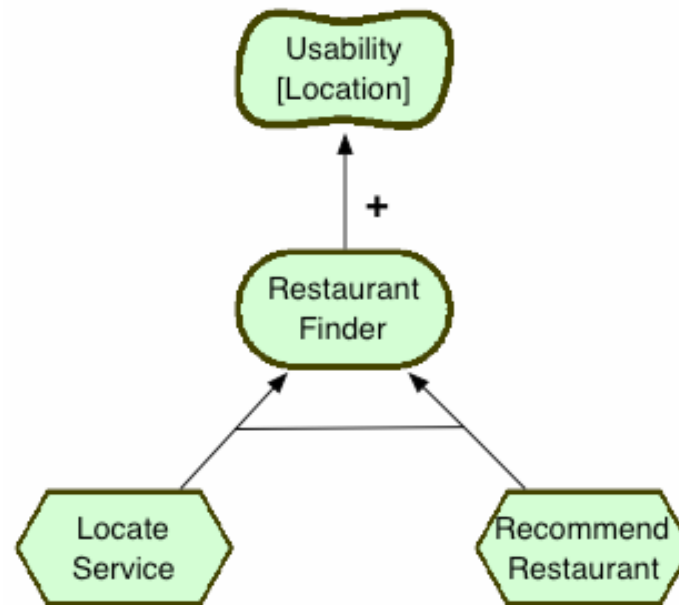
Analysis

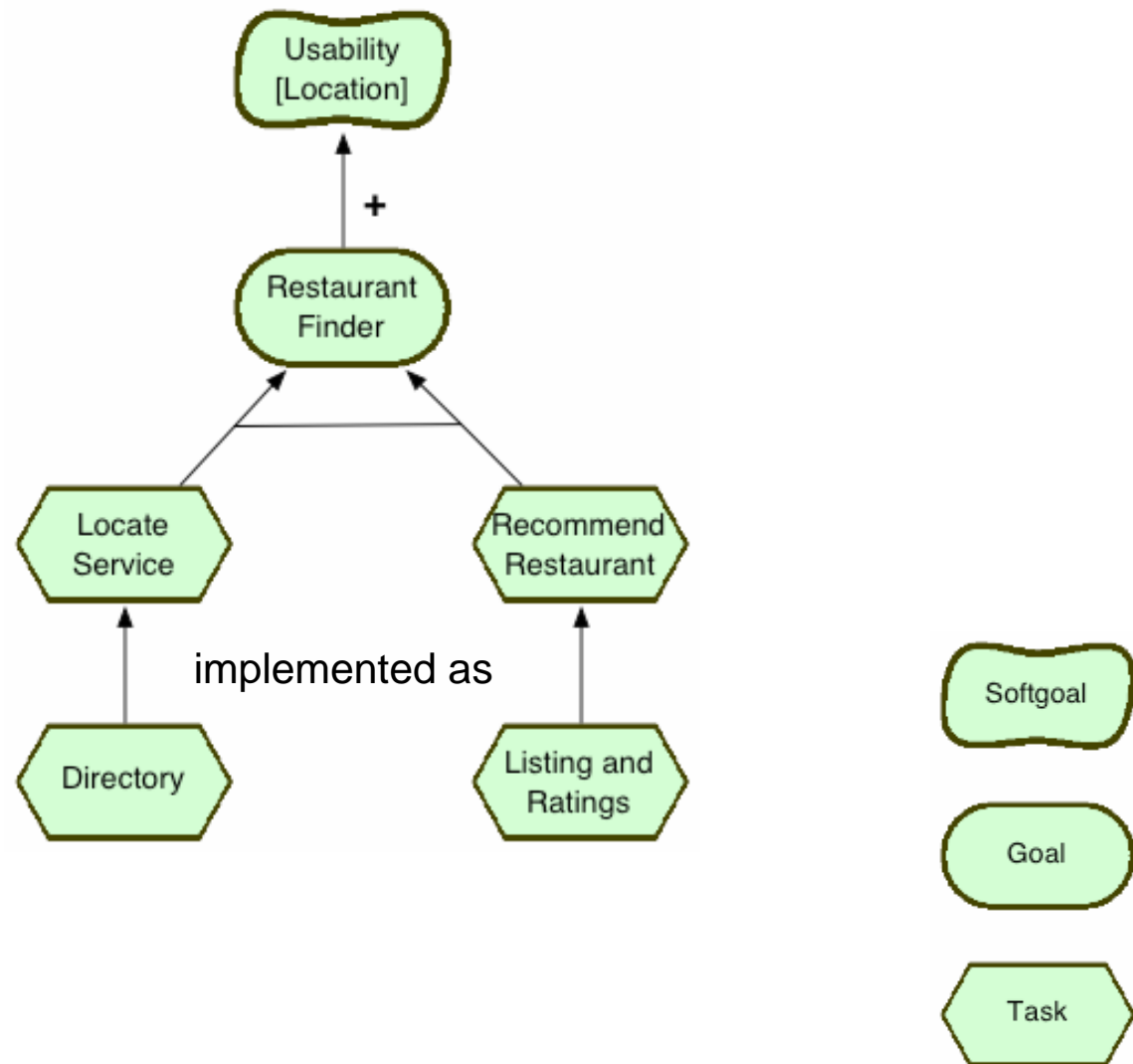
- Current identity services such as Passport blur the line between authentication and authorization
 - Identifying the user (authentication)
 - Giving site access to user profile (authorization)
-  Once user authenticates to Passport-enabled site all information is shared with the site
- Potential solution is to restrict identity service to authentication, and control access to profile within user agent (eg following the P3P proposal)

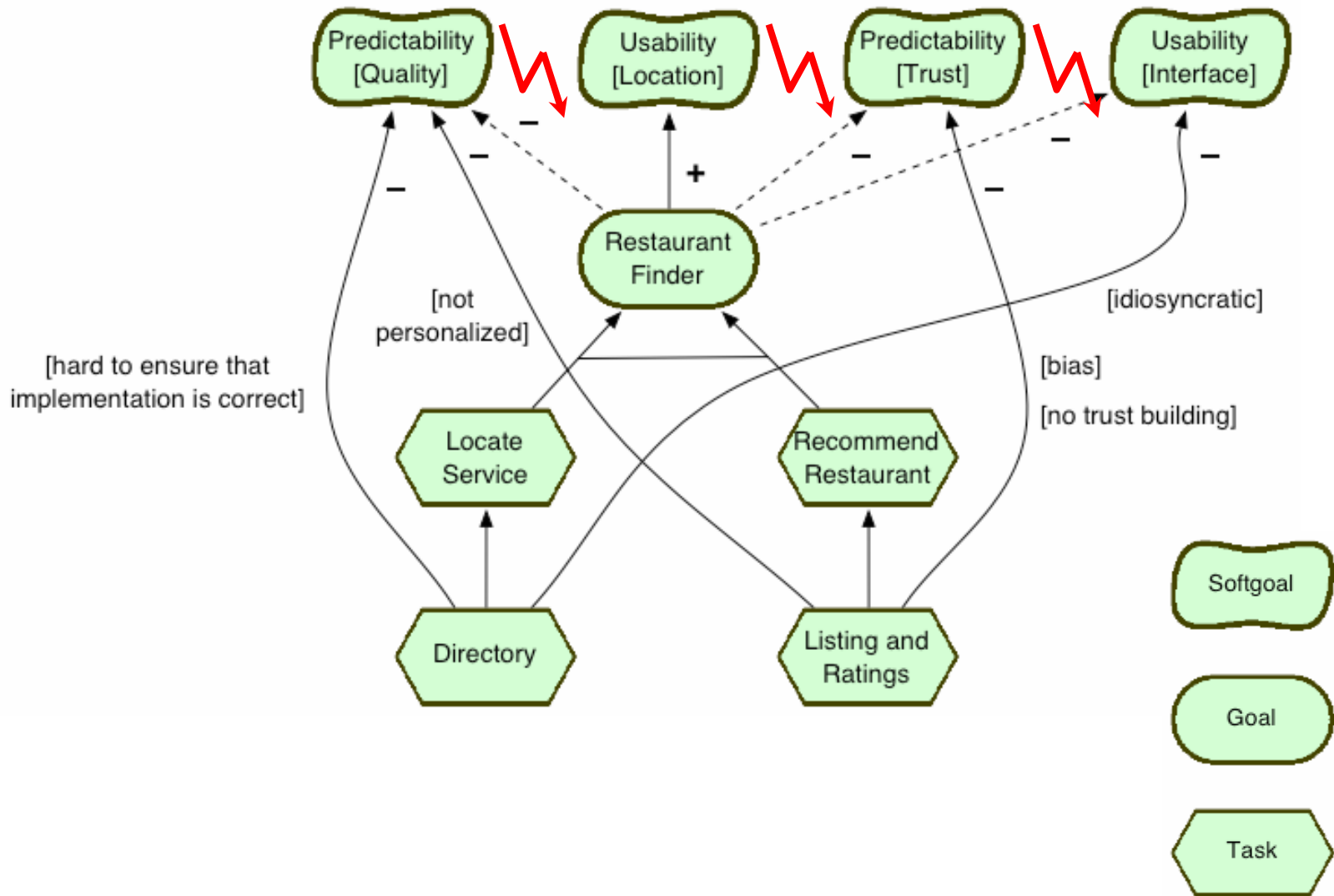
Example: Restaurant Finder

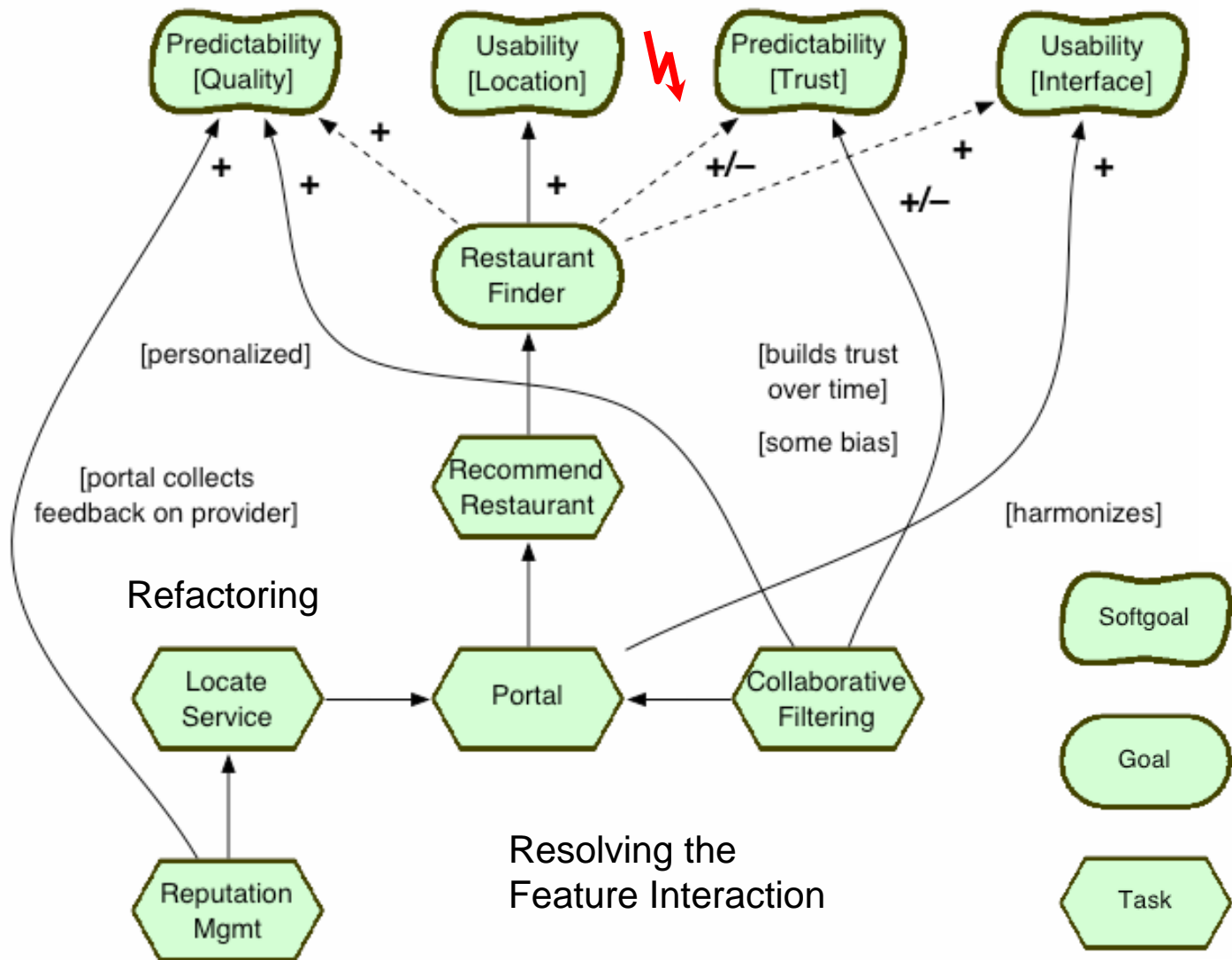
- Dynamic assembly of information services based on the user's profile & current location











Analysis

- Dynamic assembly of heterogeneous information services of unknown quality
- Issue of quality of the recommendations (eg coverage of restaurants), and bias (eg if only the restaurants that have paid a fee are listed)
- Potential solution: use trusted portal that makes the selection of localized services transparent, and polices the quality of the recommendations

Conclusion

- Feature interaction in web services
 - Functional and non-functional
- Our research on *non*-functional interactions
 - Use of GRL framework
- Goal is to develop a benchmark for FIWS
 - Set of interactions between web services
- Patterns for service composition
 - Represent NFRs as patterns