

QuanTM Architecture for Web Services

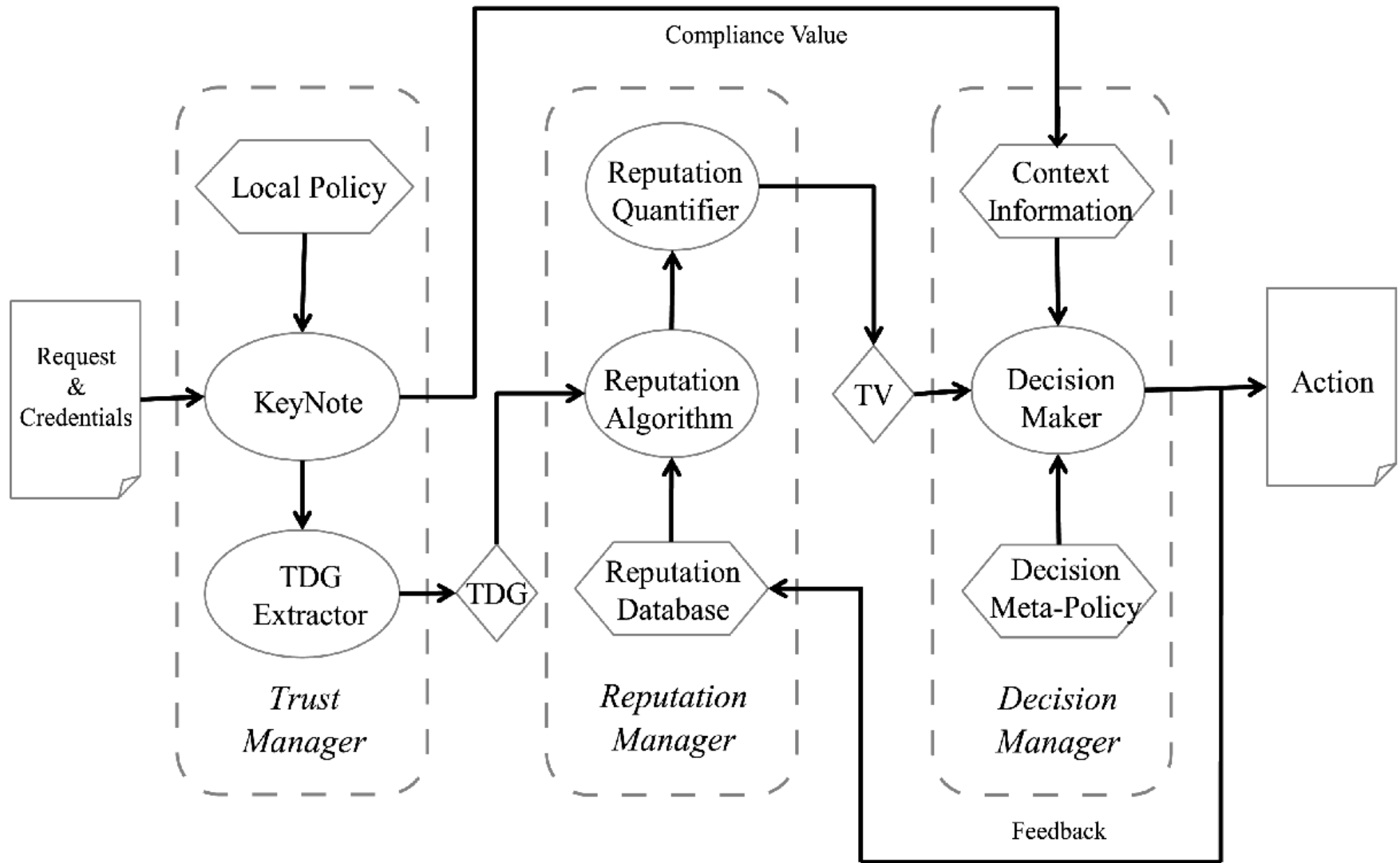
Insup Lee

Computer and Information Science
University of Pennsylvania

ONR MURI N00014-07-1-0907
Review Meeting
June 10, 2010

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 10 JUN 2010		2. REPORT TYPE		3. DATES COVERED 00-00-2010 to 00-00-2010	
4. TITLE AND SUBTITLE QuanTM Architecture for Web Services				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Pennsylvania, Computer and Information Science, 3451 Walnut St, Philadelphia, PA, 19104				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES MURI Review, June 2010. U.S. Government or Federal Rights License					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Same as Report (SAR)	18. NUMBER OF PAGES 24	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

QuanTM Architecture

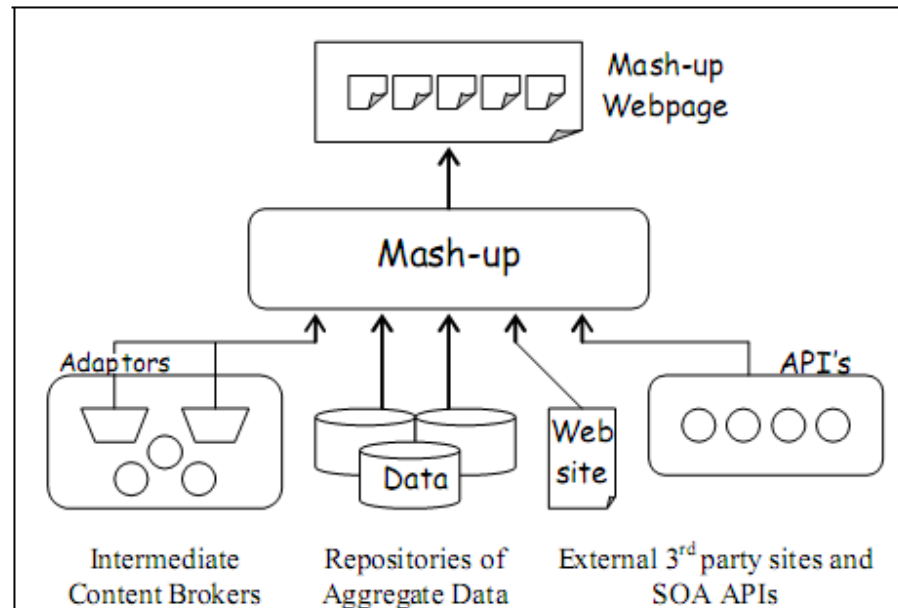


Application Area: Service Mashups

- What is Mashup:
 - Wikipedia definition
 - A mashup is a web page or application that uses or combines data or functionality from **two or many more** external sources to create a new service.
 - Definition from academia literature
 - A mashup is a website or web application that seamlessly combines content from **more than one** source into an integrated experience.
- Key aspect:
 - It involves multiple administrative/trust domains

Service Mashup

- Mashup Architecture



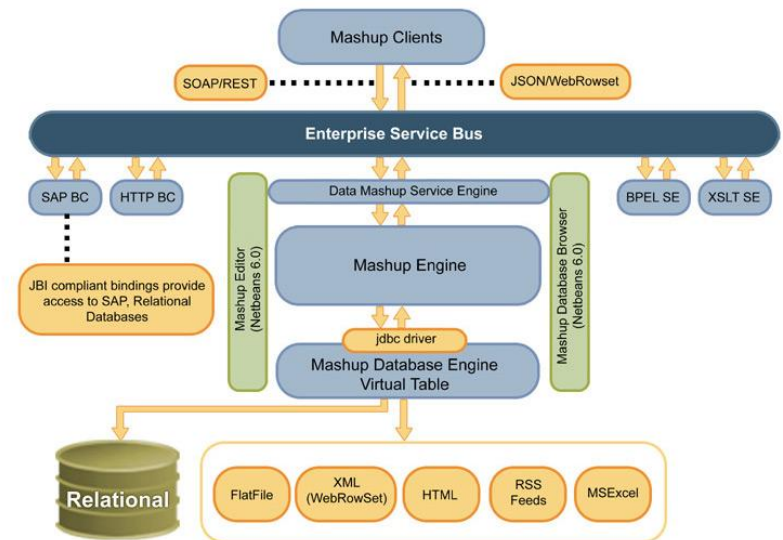
- For example, the content may be drawn from local data repositories, from existing local and external web pages, accessed via SOA based APIs, and from intermediate content brokers.

More on Mashup (cont.)

- Mashup Types:
 - Data mashups
 - combine **similar types** of media and information from multiple sources into a single representation
 - Consumer mashups
 - combines **different data types**. Generally visual elements and data from multiple sources
 - Business mashups
 - generally define applications that combine own resources, application and data, with other external web services, allowing for collaborative action among businesses and developers

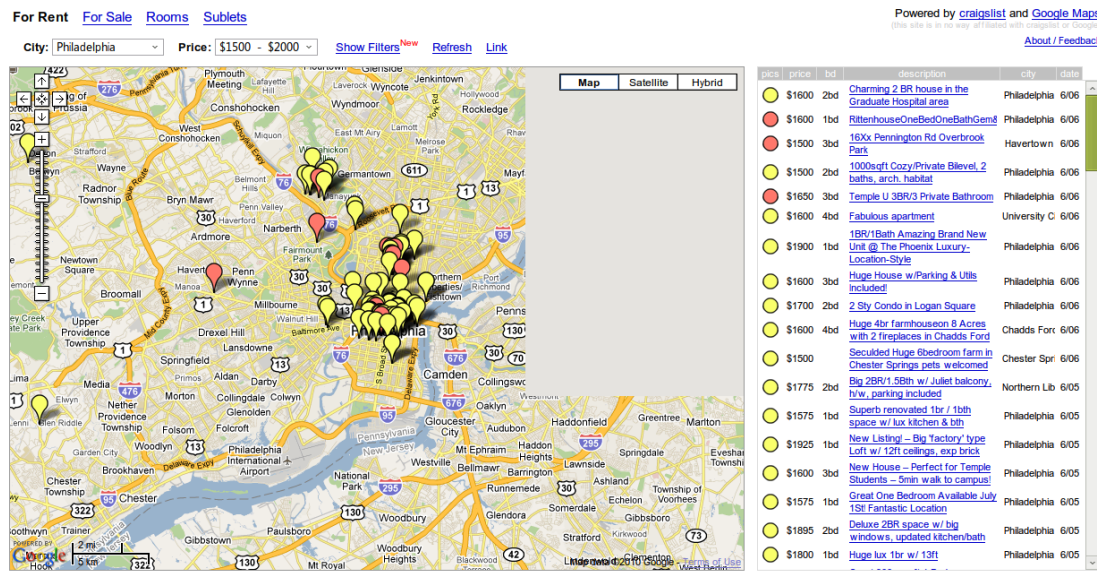
Data Mashup Example

- RSS Feed
 - Integrate new post on from various blogs, websites using Google Reader
 - Integrate headline news from various news source, such as: NY-Times.com, CNN.com, and BBC.com
- Enterprise Data Mashup
 - aggregate relational datastores represented as federated query server



Client Mashup Example

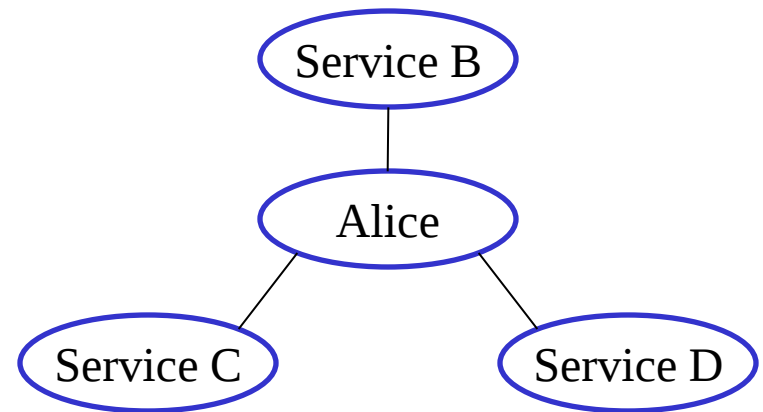
- One widely-cited example of web mashup:
 - www.housingmaps.com combines Google Maps data with Craigslist's housing data and presents an integrated view of the prices of the houses at various locations on the Google map.



Business Mashup Example

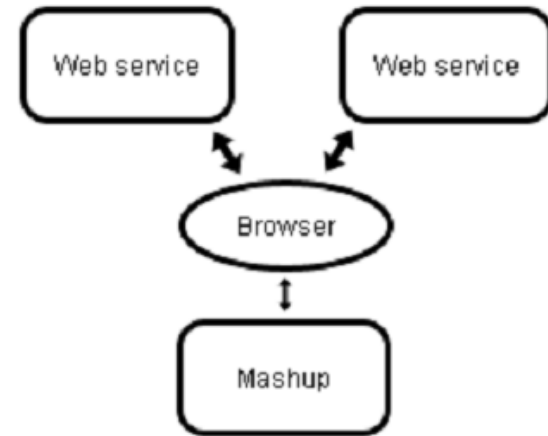
- E-trading mashup is a trading platform to allow their customers to trade globally.
- For a particular trading transaction:
 - Customer Alice initiates the trade request with Service B.
 - This is based on the pricing chart provided by Service C's charting service, with real-time price input from Service D.

Identity	Role
Alice	Trade Requestor
Service B	Trade Provider
Service C	Charting Service Provider
Service D	Real-time Price Provider



Web-based Mashup (WbM)

- Definition:
 - mashup typically use the user's Web browser to combine and reformat the data
- Challenges:
 - Need to transfer/share information cross multiple trust domains
- Limitations:
 - The current security model used by web browsers, the Same Origin Policy (SOP), does not support secure cross-domain communication desired by web mashup developers.
 - The developers need to choose between:
 - **no trust**: where no cross-site communication is allowed
 - **full trust**: where third-party content runs with the full privilege of the integrator (mashup provider), after explicit user consent



Server-based Mashup (SbM)

- Definition:

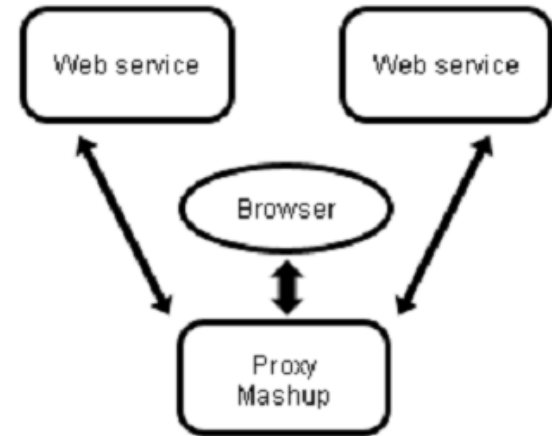
- analyze and reformat the data on a remote server and transmit the data to the user's browser in its final form

- Features:

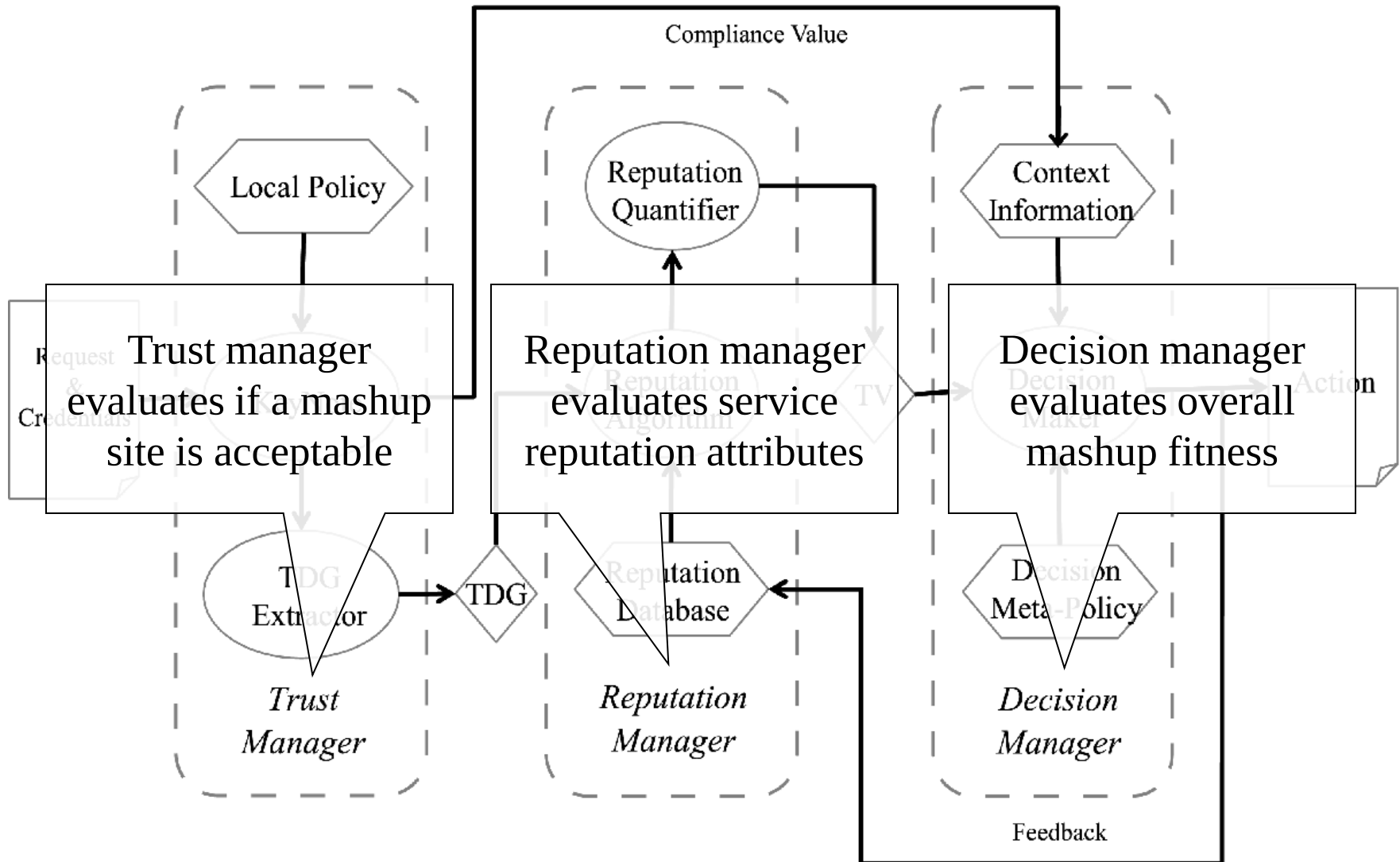
- It does not suffer from the SOP limitation
- Security issues can be addressed using corresponding security protocols/standards, such as: OAuth authentication technique

- Limitations:

- Requires user to **give complete trust** to mashup providers on accessing his/her private data
- Need a **proxy mashup service** instead of using client-side computation resource.



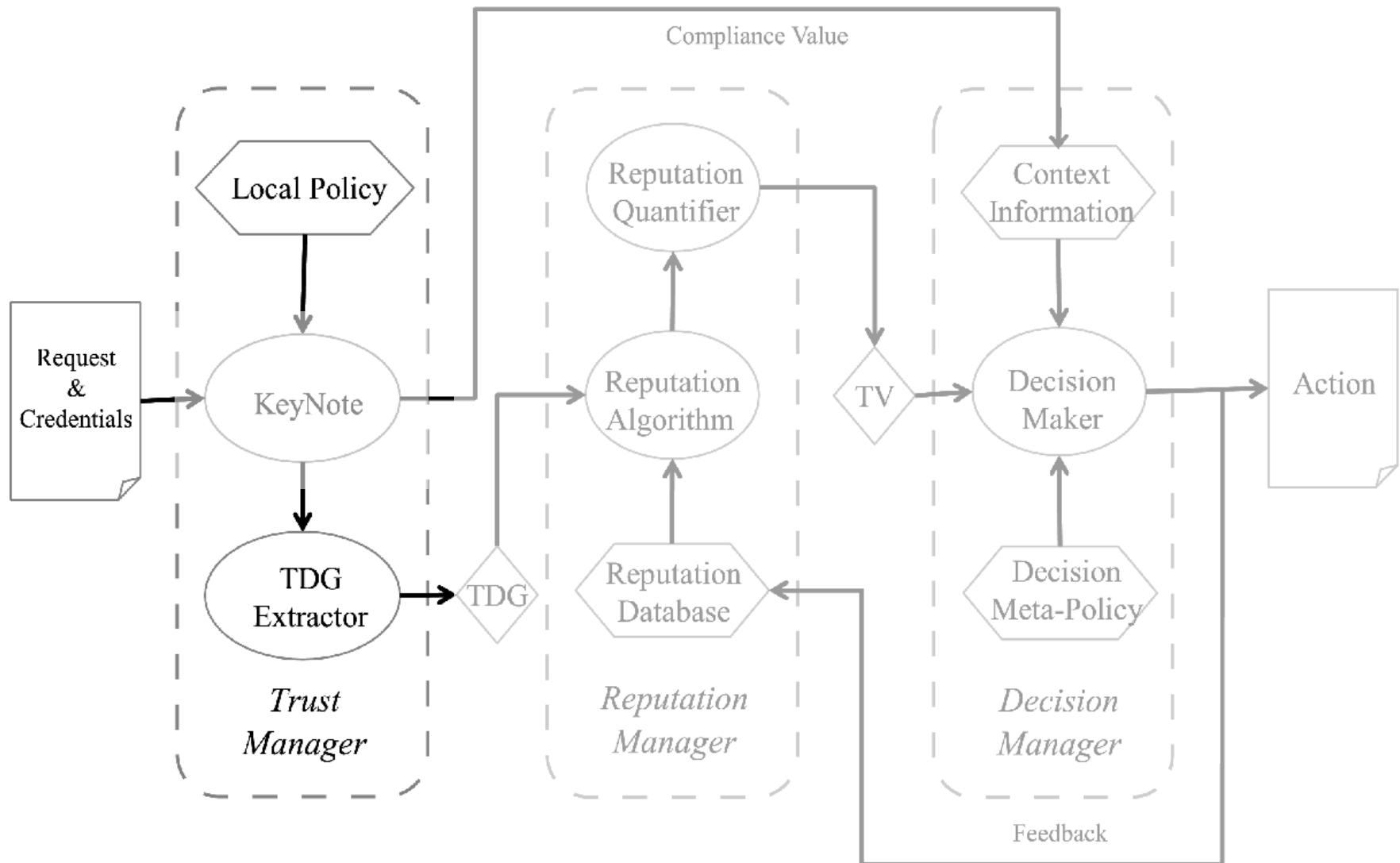
QuanTM to the rescue!



Which mashup to use/trust?

- QuanTM can be applied as a powerful trust framework, replacing current SOP
- Some mashups may be ruled out by *local policy*.
 - *E.g.,: no mashup is allowed to execute, if:*
 - some service components do not support secure connections
 - it needs third-party service to read user email contacts
- Acceptable mashup according to static local policy may have various trust-levels:
 - *E.g., one service component is known to leak user information to third-party violating privacy requirement*
- *Decision policy* used to make the final decision
 - Different from the aforementioned local policy above

Trust Manager

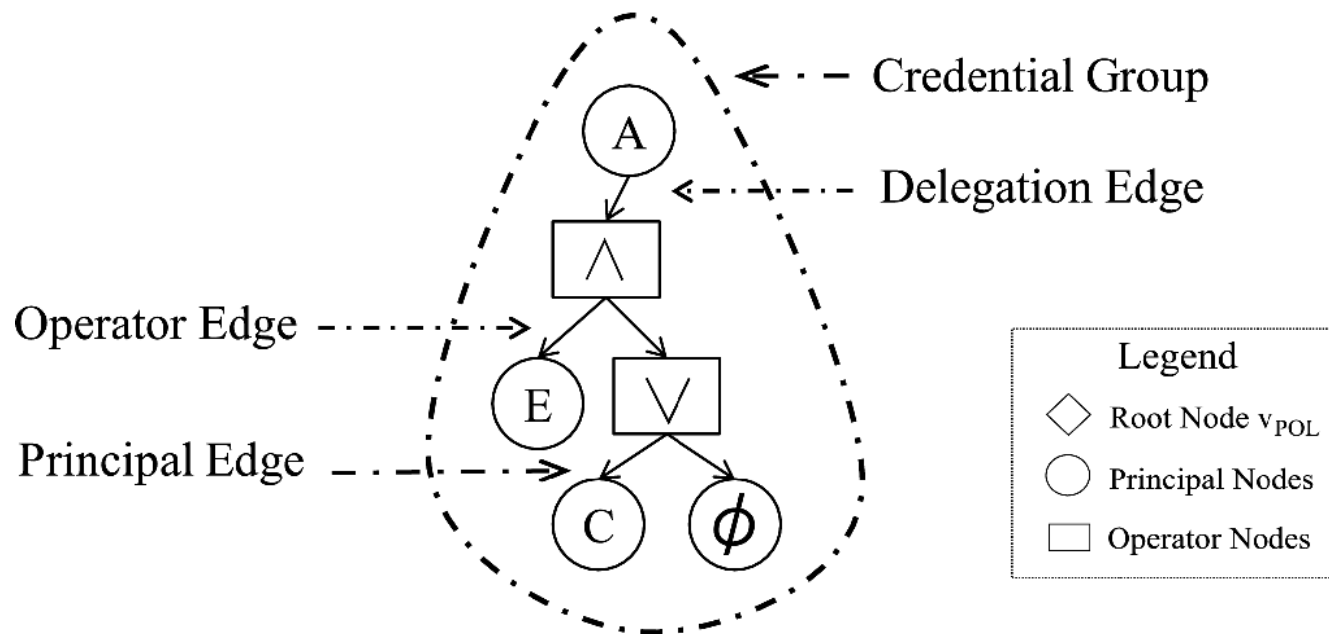


Trust Manager

- Identify service components of mashup
 - DNS name as identifier
- Use local policy to evaluate mashup compliance
 - Qualitative value representing compliance with the policy
- Construct trust dependency graph (TDG)
 - Based on mashup dataflow and workflow
 - May require analyzing underlying javascript code

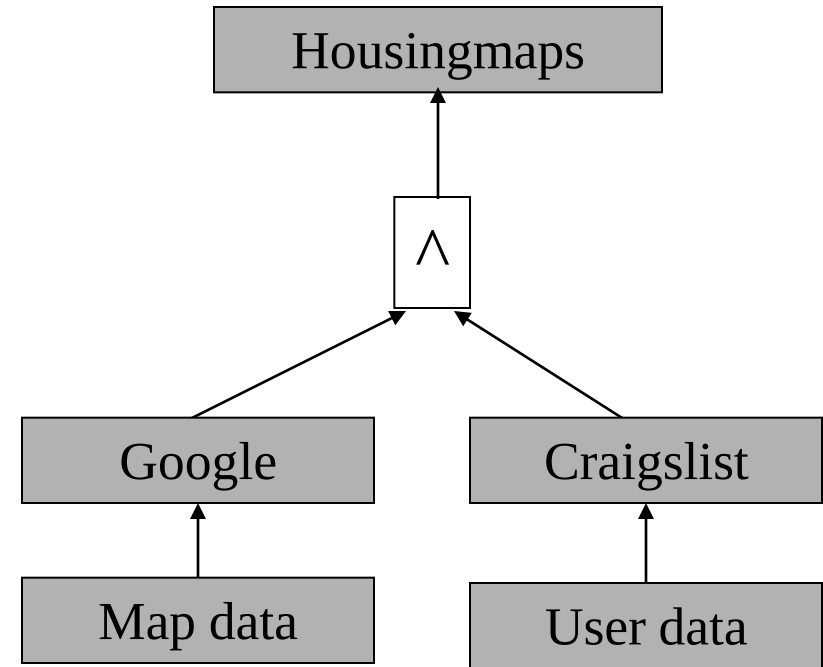
Trust Dependency Graph (TDG)

- An encoding of mashup workflow/dataflow
- Reflect trust in principals and trust relations
- Edges represent trust dependencies
- Reputations are assigned to TDG elements

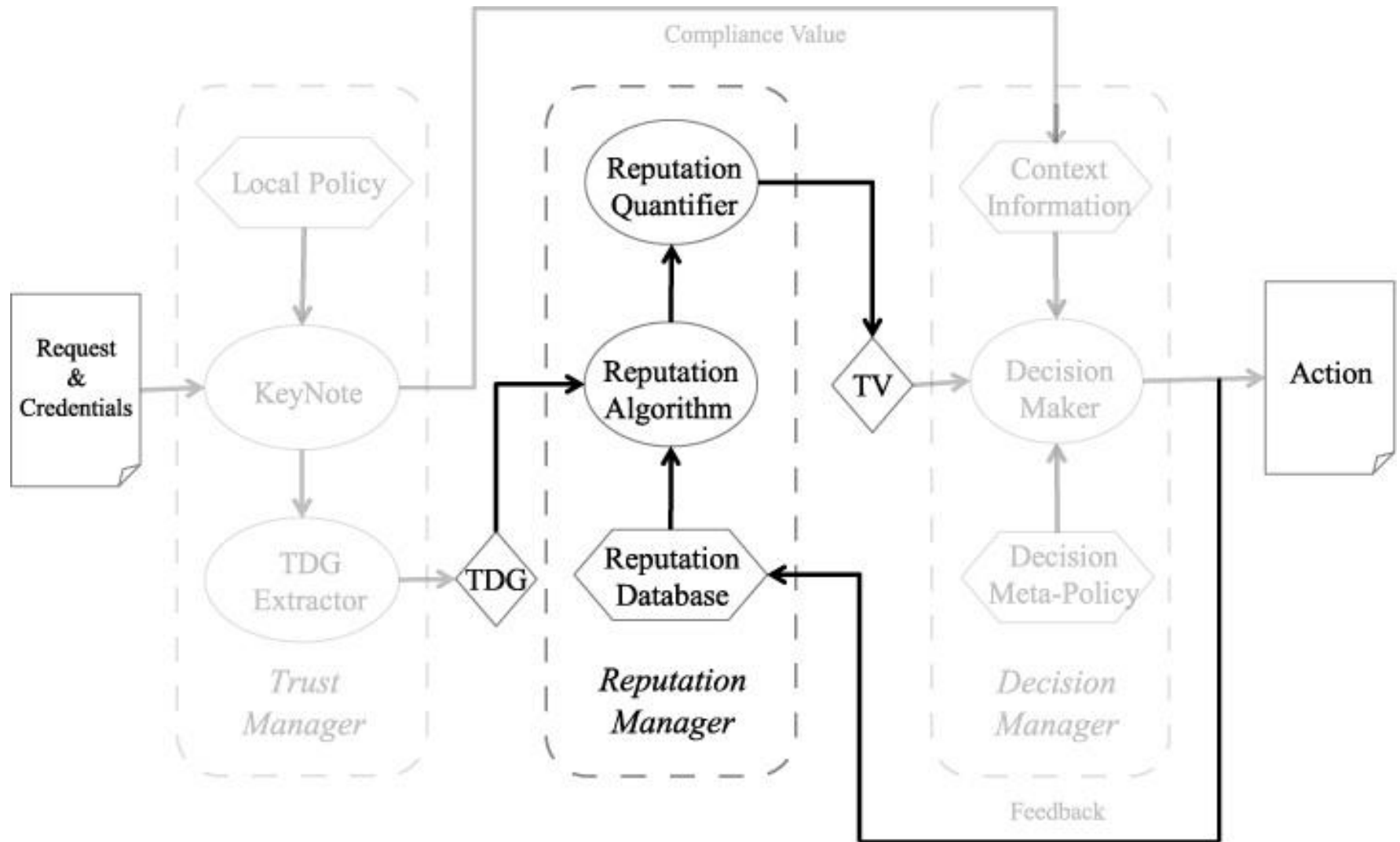


TDG for Housingmaps

- Data sources:
 - map data
 - house listings
- Services:
 - Google
 - overlay data on maps
 - Craigslist
 - deliver user data
 - Housingmaps
 - parse and filter data from Craigslist
 - send to Google
 - arrange results



Reputation Manager

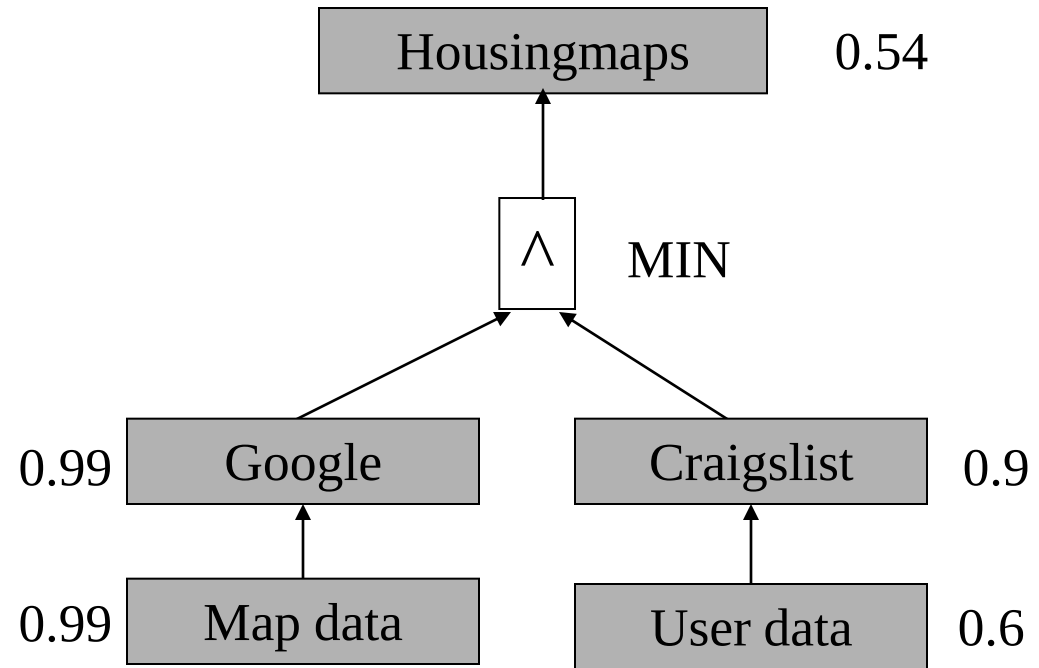


Reputation Manager

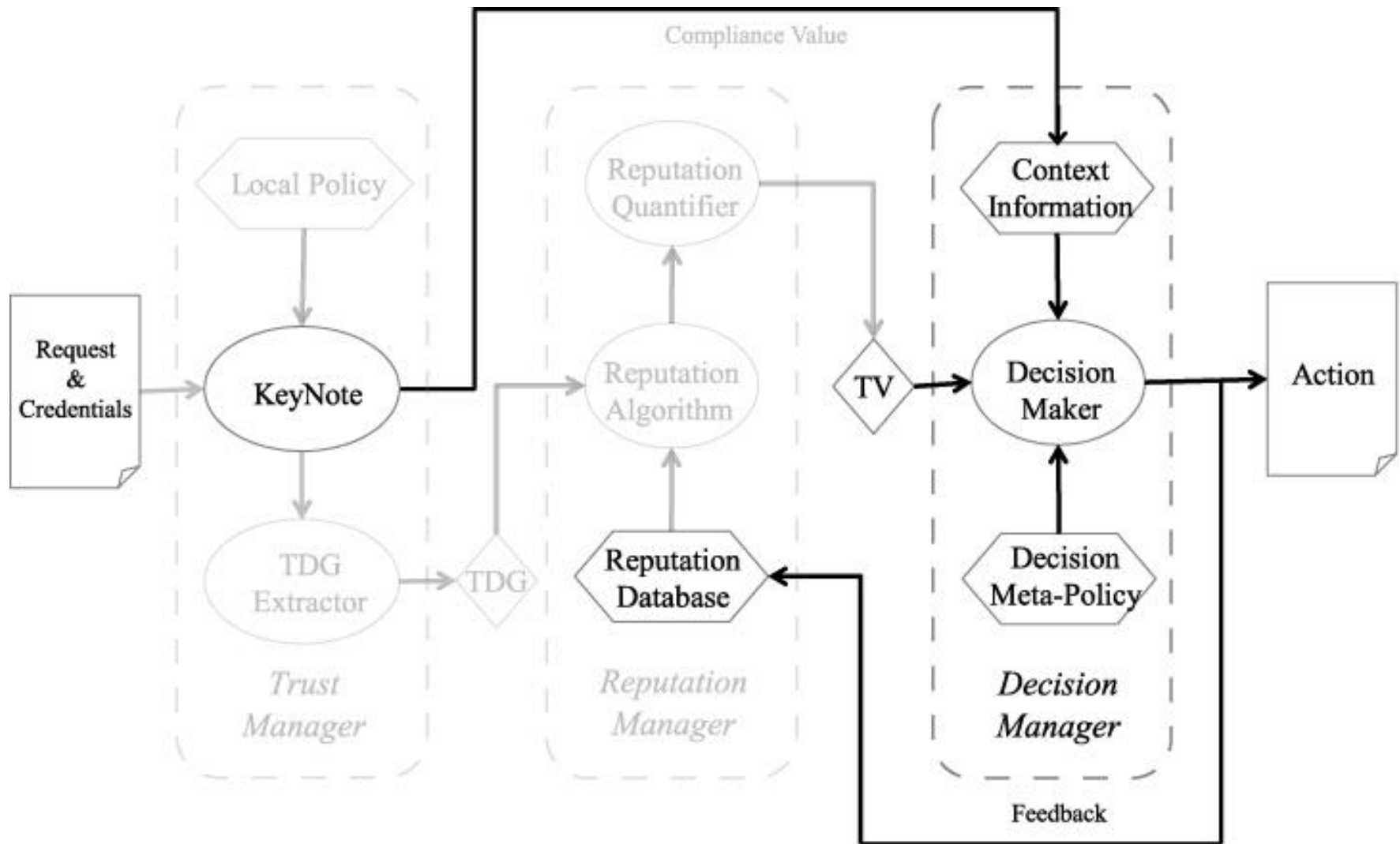
- Calculate trust value
 - Assign reputations to TDG edges using reputation values
 - “push” reputation values up the graph
- Build reputation using existing databases, e.g.,
 - General DNS reputation DB
 - Google PageRank
 - Trust-of-Web reputation DB
- Update reputation based on feedback
 - Past performance of the service
 - Experiences from other mashups and direct uses

TDG for Housingmaps with reputation

- Data sources:
 - map data
 - house listings
- Services:
 - Google
 - overlay data on maps
 - Craigslist
 - deliver user data
 - Housingmaps
 - parse and filter data from Craigslist
 - send to Google
 - arrange results



Decision Manager



Decision Manager

- Uses an user-specific meta-policy
 - Context monitors
 - Cost-benefit analysis
 - Game-theoretic formalization
- Simple example: Threshold policy
 - If $CV='maybe'$ and $TV>0.5 \rightarrow$ Fulfill request
 - If $CV='true'$ always fulfill request
 - In general, thresholds can be adaptive

Current and Future Work

- Design local policy language for WbM
 - Allow user to specify their static trust requirement
- Technique to construct TDG for WbM based on code (e.g., javascript) analysis
- Integrate available service reputation
 - E.g., DNS reputation, PageRank, Trust-of-Web Score
- Design Decision policy language for WbM
 - Allow user to specify their dynamic trust requirement
- Implementation of QuanTM WbM as extension for real-world application (e.g., Firefox web browser)
- TDG-carrying services

End

Applying QTM to Mashup

- Evaluation and selection of services to use
- Differences from access control:
 - “request” is now an entry for consideration
 - Services may be evaluated initially and/or re-evaluated periodically
 - “delegation” is one service using another as part of its operation
 - “policy” describes rules for selecting and comparing services
- Similarities:
 - Trust in an entity depends on dependencies and past performance
- Issues
 - Accountability
 - Authentication (access control) and non-repudiation guarantees need to be provided.
 - Service Selection
 - Need to understand the QoS of available service components, and choose the most suitable/trustworthy ones to build mashup