Representing Context in the Mobile Internet

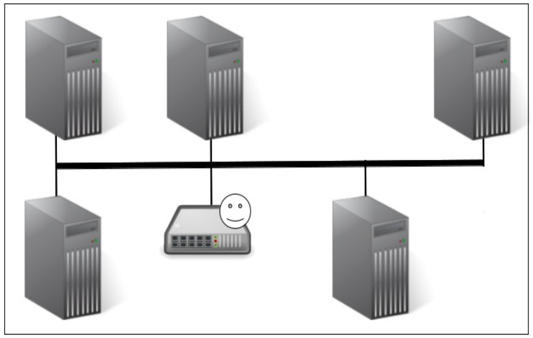
John-Austen Francisco
deymious@cs.rutgers.edu
Department of Computer Science,
Rutgers University.

Richard Martin
rmartin@cs.rutgers.edu
Department of Computer Science,
Rutgers University.

FIA: Mobility First

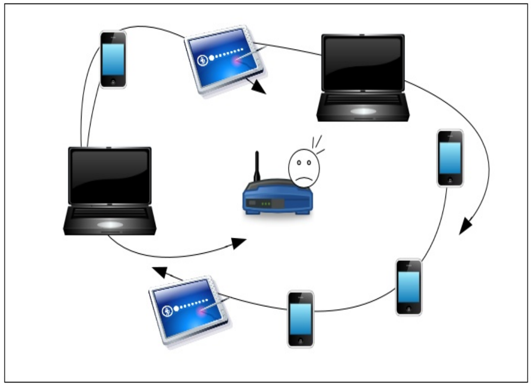
Future **I**nternet **A**rchitecture: Mobility First

- Clean-slate Internet re-engineering project
- What's so wrong with the current Internet's architecture?
- Nothing, we're using it badly



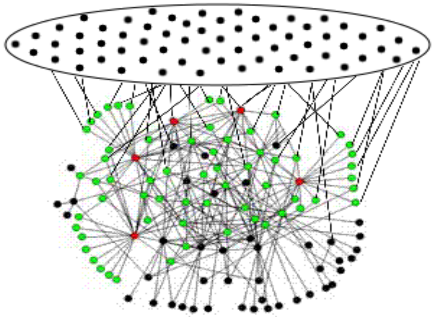
Connection-Oriented Semantic:

- network address is identity
- stateful association
- highly mobile devices
- reassociation clears connection state



Identity-Oriented Semantic:

- Abstract identity with a **G**lobally **U**nique **ID**entifier
- **G**lobal **N**ame **R**esolution **S**ervice handles GUID translation
- clients inform GNRS of their current association
- FIA routers can store client information to be delay-tolerant



RUTGERS WINLAB

Context

Contextual network communication:

- Expand and generalize on point-to-point delivery

- Send a message to an identity rather than an address
    - covered by GUID layer

- Send a message to an identity conditionally
    - send a message to whomever is in a car's passenger seat
    - send a message to a car's driver only if the car isn't moving

- Send a message to an event
    - all people at teatime
    - anyone at a talk
    - anyone who is in the building that isn't in a meeting

Context?

What is Context?

- Anyting outside the network that affects the communication
- 'outside' means the network doesn't understand it
- context service translates outside events into things the network understands

Definitions:

- **N**etwork **A**ddress
  - the routable end host address that the network can deliver to directly
  - least abstract, can be used directly by the network
- GUID
  - the FIA-routeable identifier for a given end host
  - one-step abstraction, can be translated to a NA by the GNRS
- Context Expression
  - boolean expressions on GUID attributes
  - two-step abstraction, can be resolved to GUID(s) by the context system
- Context Name
  - symbolc name for a context expression
  - third-step abstraction, references a context expression

Context!

Case for Context

- Have context systems out there now
- Why build context into the network?
- When context spans everything, by default, simple relations can be very powerful

From Examples:

- Send message to identity
    - map identity to network address

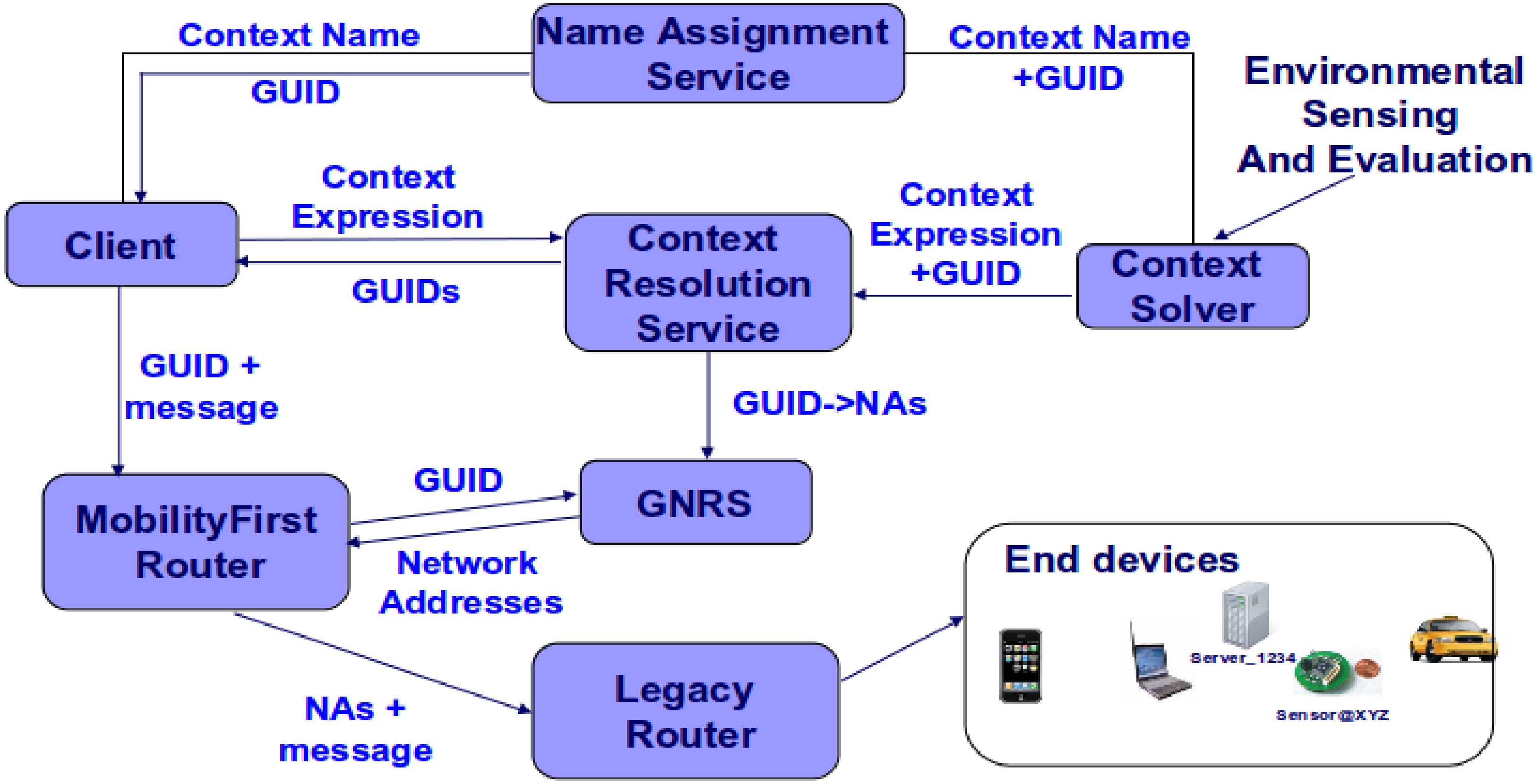
- Send message conditionally
    - sample outside data
    - if threshold is met, forward to target address

- Send message to an event
    - discover all network addresses that might satisfy the event definition
    - filter out those that do not meet the definition
    - forward to the rest

System Architecture

What is context?

- Anything outside the system that should alter the system's behavior
- context system's role is resolution and translation, not sensing



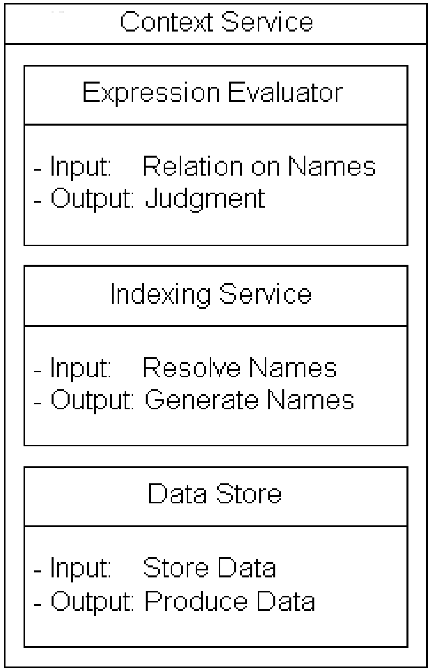
Representing Context

- Not interested in actually *representing* everything, only boolean decisions on it
    - did event X happen?
    - is data X within bound Y?

- Splits the world into things we know how to compare or don't
    - enumerate unknown things
    - enumeration closes equality
    - reference enumeration order to close comparison

    e.g.: color = {'red', 'green', 'blue'}

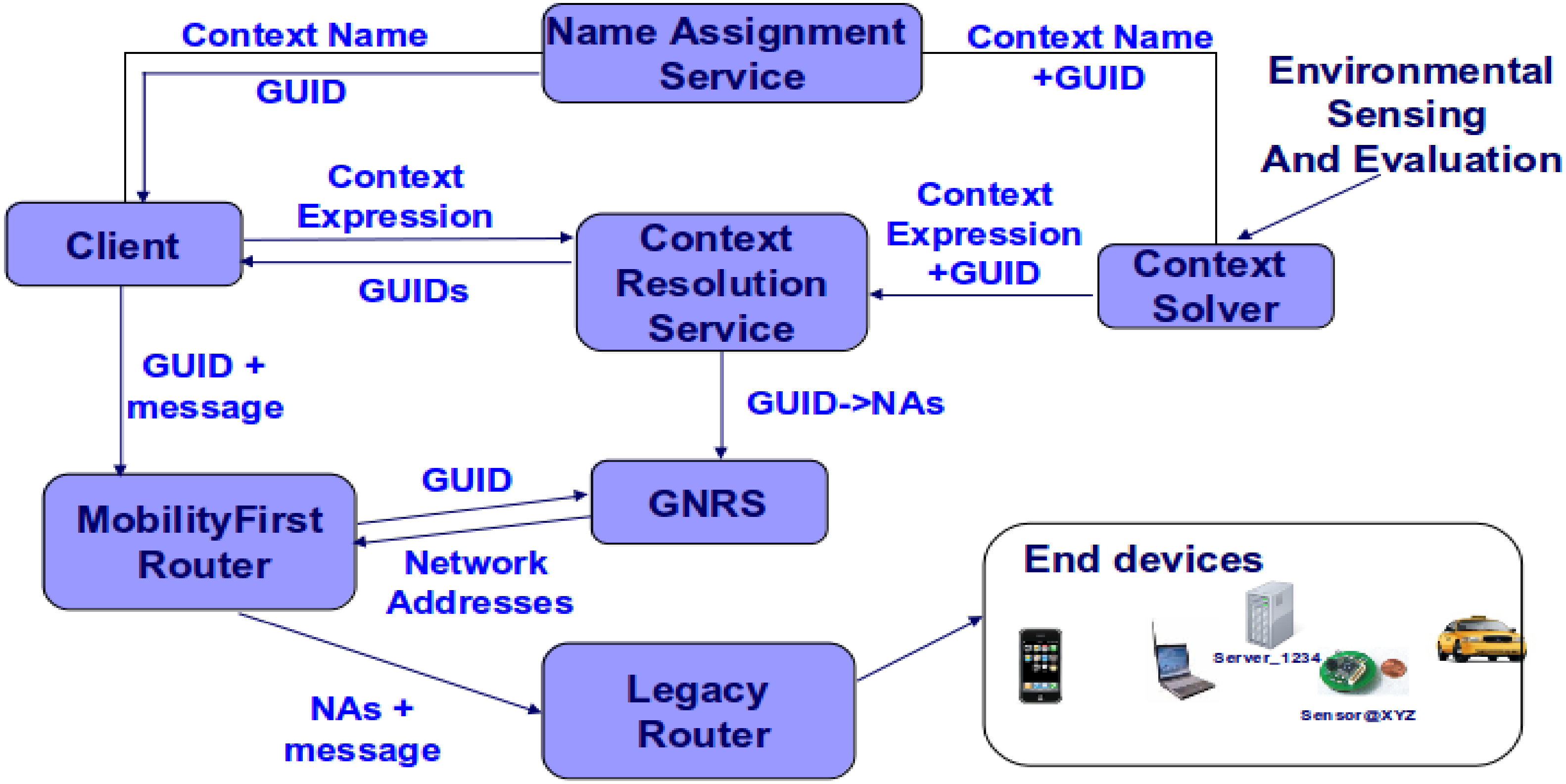
- Context samples as triples:
    - <GUID, attribute name, attribute value>



Context Examples

Using context:

1. Send to a Name
2. Send to a Location
3. Condtional Send



RUTGERS

WINLAB

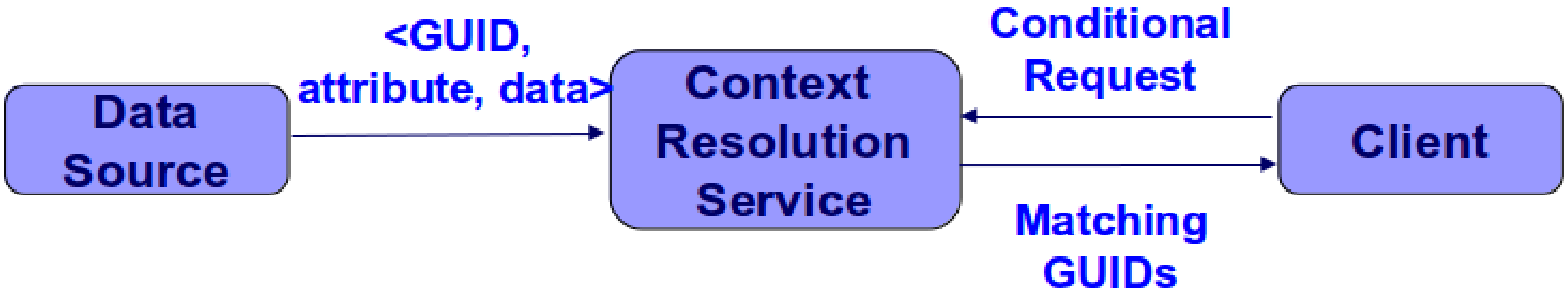
Early Resolution

The Client knows how to get what they want

- Client sends condition request
- CRS resolves it based on sensed-data triples
- CRS gives back all matching GUIDs

Examples:

- Send to a Name → GUID where identity attribute = X
- Send to a Location → GUIDs where location attribute is bounded by expression Y



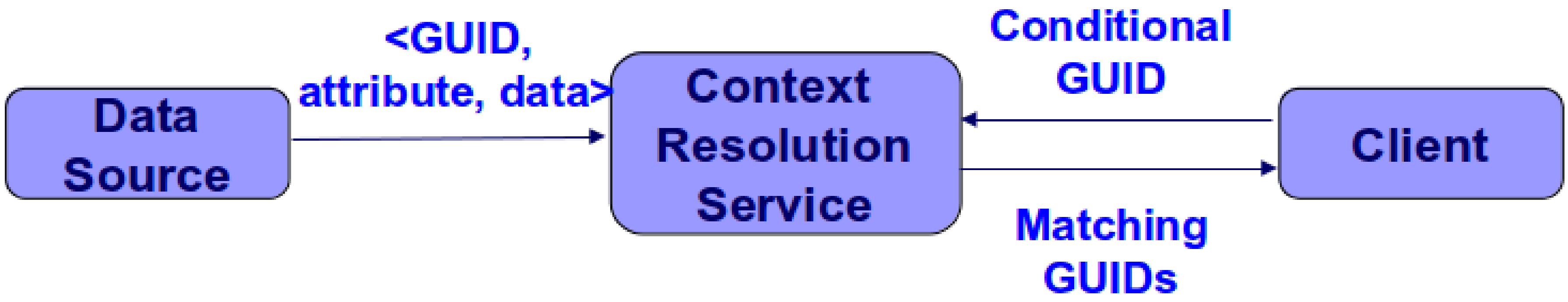
On-Time Resolution

The Client knows the name of what they want

- Client sends conditional GUID
- CRS resolves it based on sensed-data triples
- CRS gives back all matching GUIDs

Examples:

- Send to a non-driver → GUID of person if they aren't driving, GUID doesn't change
- Send to a place's GUID → GUIDs of all in that place



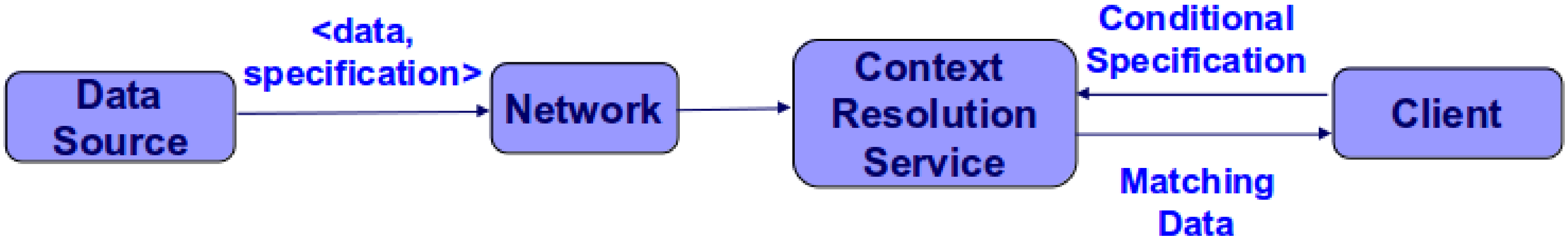
Late Resolution

The Client knows how to describe what they want

- Client sends conditional expression to CRS
- Data Source injects described data/content with no destination
- CRS resolves data targets based on expressions
- CRS relays data/GUIDs to client

Examples:

- Fetch Election returns → All things that match contextual specification
- World news noon to 1 → All updates in a certain time range
- Who tried to contact me 3am to 8am → Basic away caller-ID-like function



Generalizing Delivery

Context on GUIDs

- GUIDs are general enough to mean anything
- Clients do not need to understand them
- We can alias GUIDs to things that aren't network objects

What does this buy us?

- generalized delivery
    - can deliver to target based on:
        - identity
        - contextual conditions the client knows
        - contextual conditions the target stipulates
        - contextual specification the client imposes
    - can alias a GUID to a context expression
        - on 'communicating' with the GUID, the expression is resolved
    - can create network-level names for things that are not objects
        - can assign a name (GUID) to expression for 'all people at teatime'
        - can assign a name (GUID) to expression for
          'everyone who called while I was asleep'

Questions?

not context free:
S->A|AS
A->a|B|C
B->b
aCb->acb

context free:
S->AS
A->a|B|C
B->b
C->c

non-free:
A can go to C, but only aC goes to a terminal, so must have AA->aA->aC->ac
A must go to a and B must go to b before C can be resolved to c. This grammar restricts the order of the parse tree.

fundemental difference is that nonterminals and terminals are grouped in some way. Terminals have no ambiguity, they can not be parsed down to anything else. A terminal grouped with a non-terminal partially cements the parse tree for the non-terminal.

Future **I**nternet **A**rchitecture

- Clean-slate Internet re-engineering project
- Standard Computer Science solution; another layer of abstraction

Identity-Oriented *Context* Semantic(?)

- GUID overlay alters the landscape considerably
- GUIDs form a context-free grammar on the network
- IP subnetting and CIDR masking create hidden correlations that often can't be discovered or modeled

not context free	context free
S → A|AS	S → A|AS
A → a|B|C	A → a|B|C
B → b	B → b
aCb → acb	C → c

productions	type
aaa	both
acb	both
cac	context-free only
bc	context-free only

TEMPLATE

Future Internet Architecture

- Clean-slate Internet re-engineering project
- What's so wrong with the current Internet's architecture?