Distributed Architecture Definition Language
DADL

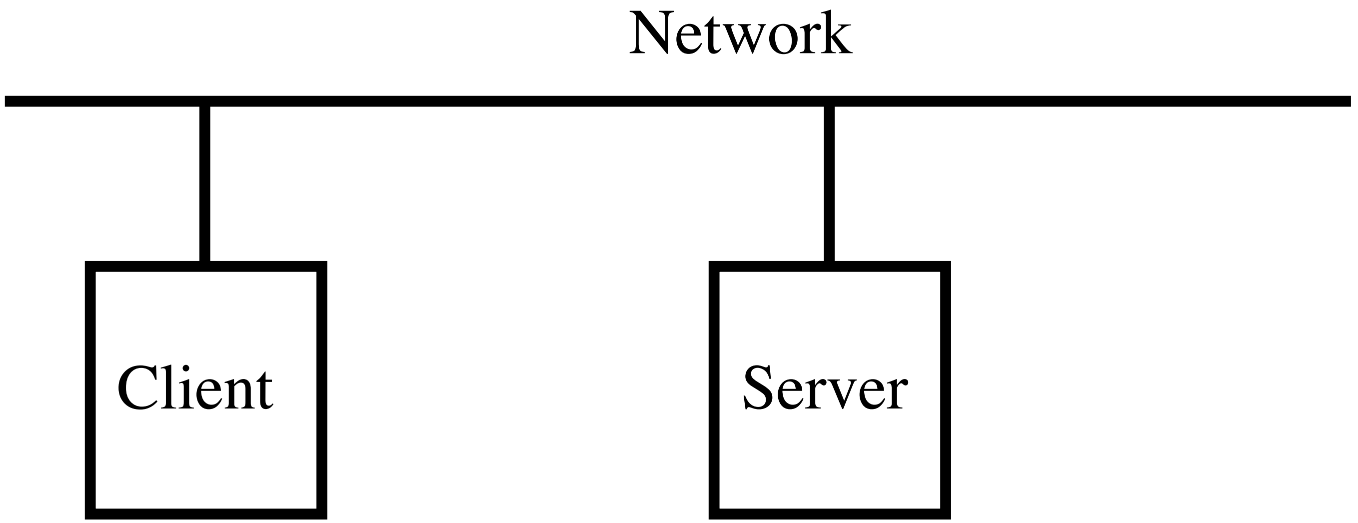
Ron Burback

October, 1997

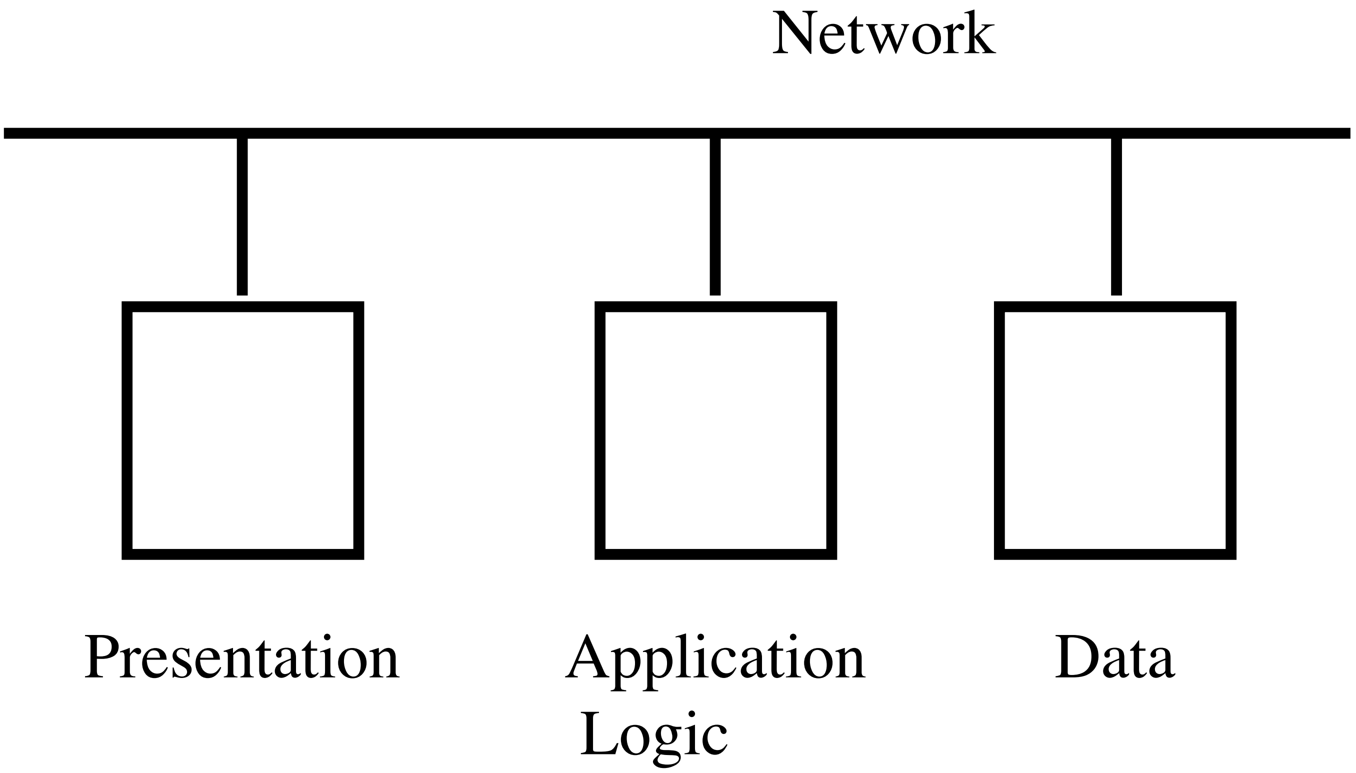
Areas of Focus

Control	plan repair, re-planning, process changes, plan optimization, chronic problem management, ...
Measure	number of faults both reported and fixed, lines of code, closeness to plan, resource utilization, performance, ...
Strategies	methodologies, architecture, paradigms, mission, risk analysis, scheduling, priority setting, resource utilization, decision making, life cycle management, ...
Tools	compilers, debuggers, environments, quality assurance, CASE, version control, databases, operating systems, networks, file systems, GUI builders, composition, ...
People	group interactions, skill development, group dynamics, communications, goal setting, ...

Basic Client/Server



Three Tiered Client/Server



Network

Presentation

Application Logic

Data

Client/Server Network Placement

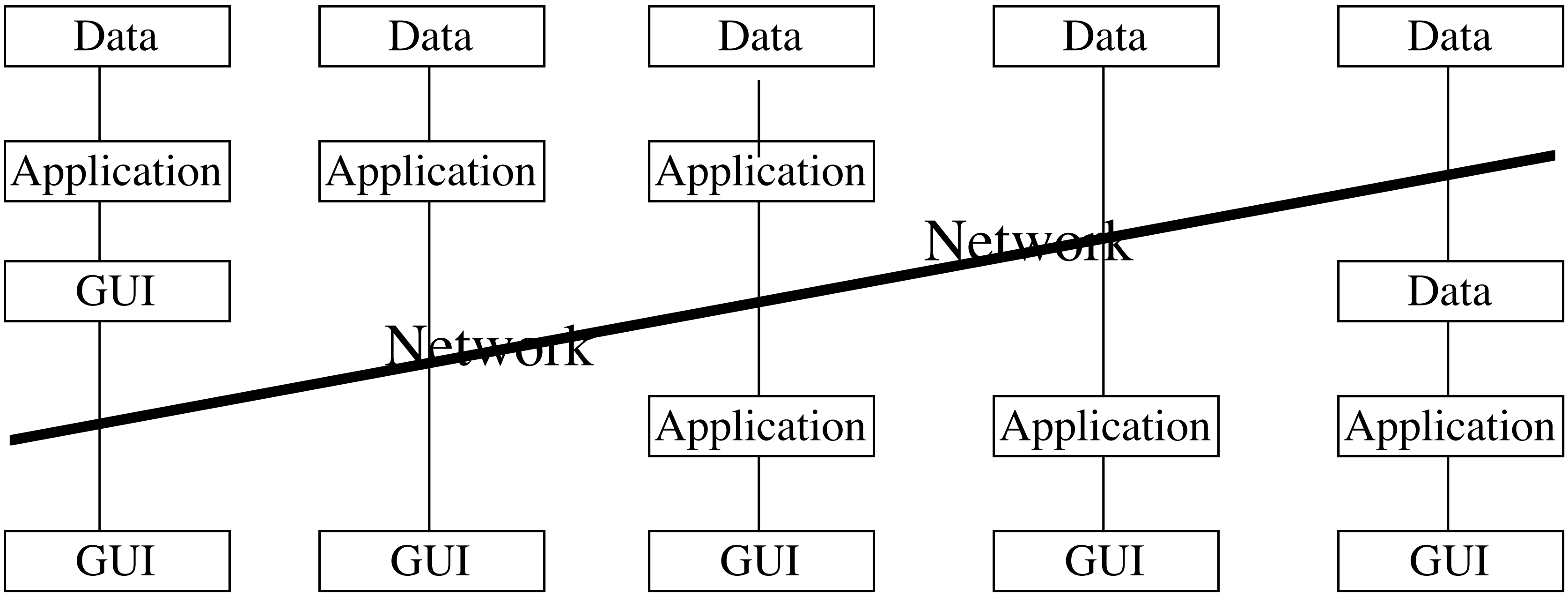
Distributed GUI

Remote Presentation

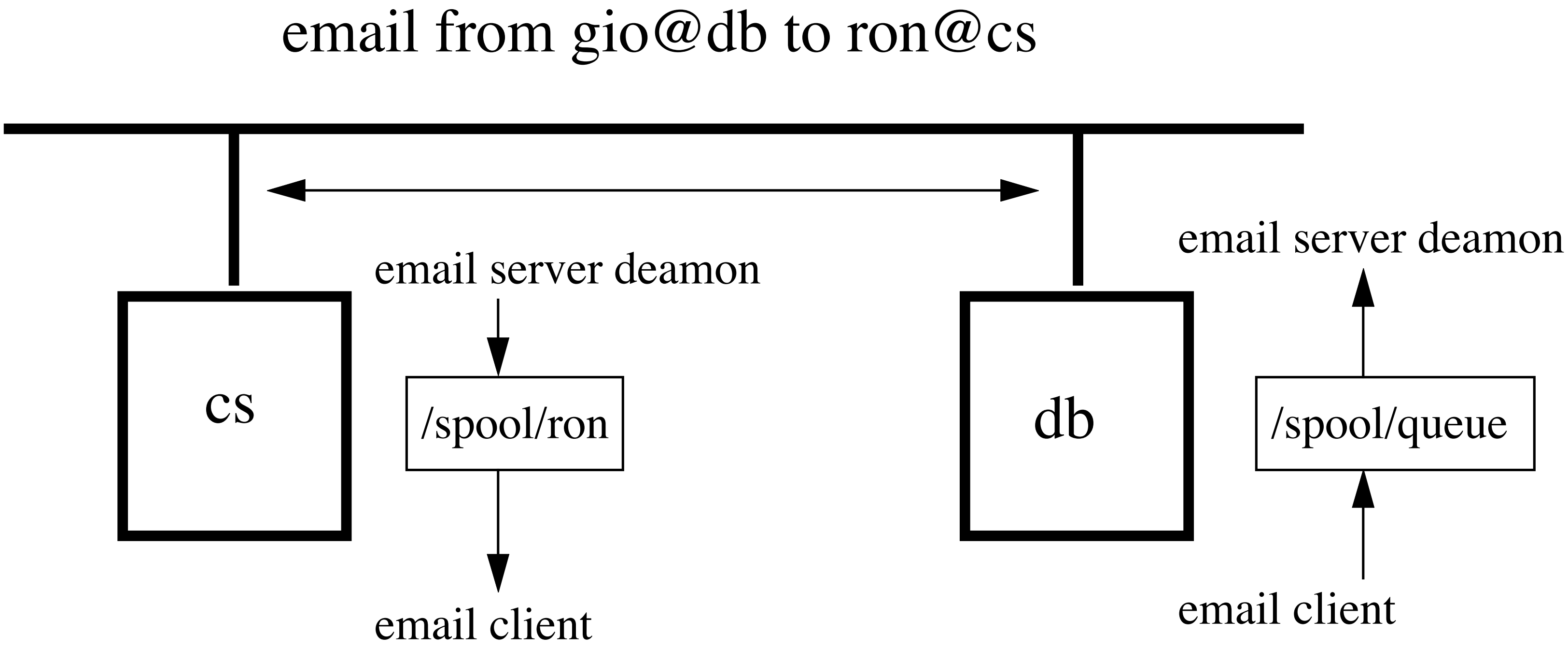
Distributed Application

Remote Data

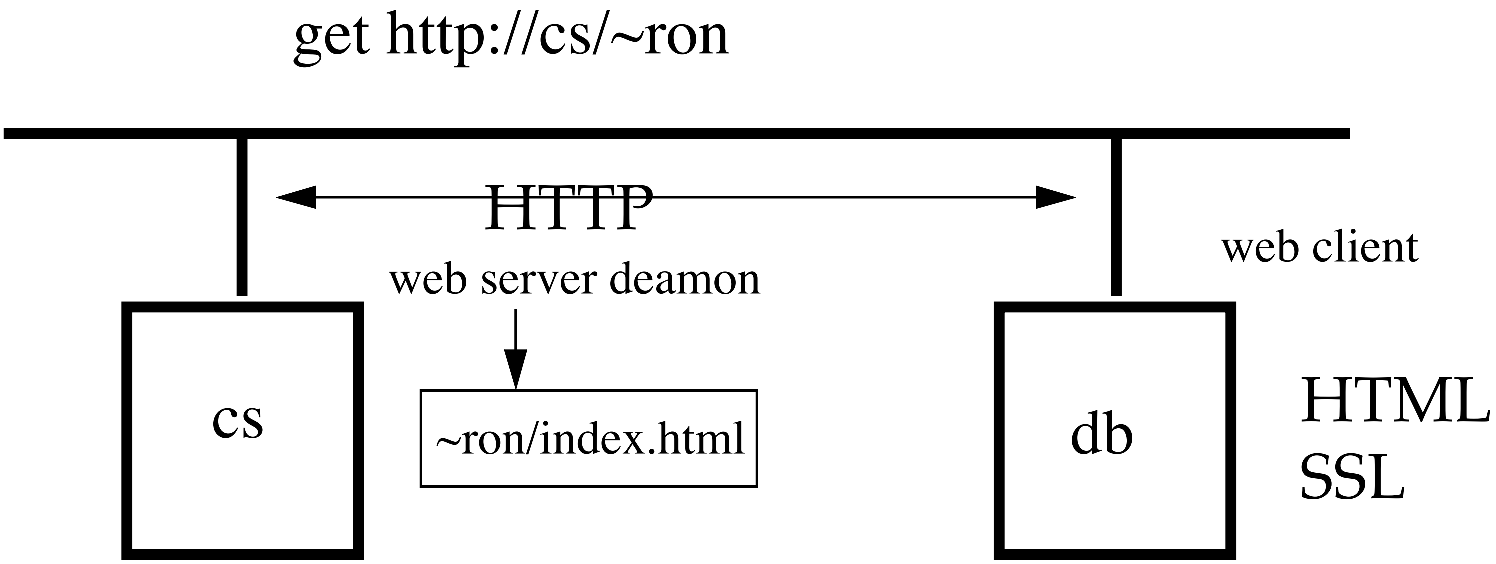
Distributed Data



SMTP based Electronic Email



HTTP based Web
Example of Client/Server



Defining Characteristics

- Distribution
- Concurrent
- No Global State
- No Global Clock
- Partial Failures
- Asynchronous Communication
- Distributed Control

- ◆ Heterogeneous Systems
- ◆ Local Autonomy
- ◆ Evolution Programming Paradigm
- ◆ Constant Change
- ◆ Many Transparencies
- ◆ Openness
- ◆ Interdependence
- ◆ Security

What is the Problem?

- Architectures are defined with a few drawings, hand waving arguments, and English based document
- Not precisely defined
- Traditional programming languages concentrate on algorithms and data structures

The Solution?

- We need a language that can describe distributed architectures.
- Extension to existing programming languages.
- DALD (Distributed Architecture Definition Language)

Architecture Definition

- The components and their interfaces, communication, and contractional behavior.
- Traditional programming languages concentrate on algorithms and data structures which define the components but do very little at defining interfaces, communication, and contractional behavior.

Consider this simple program which adds two integers.

- void main () { int results = plus(2,1) ; }
- int plus (int n, int m) { return n+m ; }

What is the architecture?

There is an implicit architecture, so implicit that it
is seldom mentioned. The two components communicate
using a shared address space and a call stack frame. The
communication is assumed error free and both components
are flawless.

Comparison of Architectures

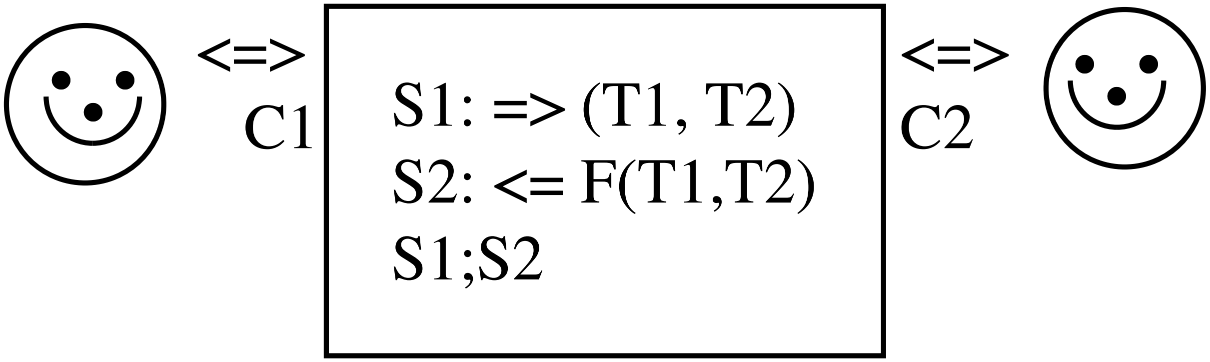
- Traditional
  - Not Distribution
  - Not Concurrent
  - Global State
  - Global Clock
  - No Failures
  - Synchronous Communication
  - Centralized Control
- Distributed
  - Distribution
  - Concurrent
  - No Global State
  - No Global Clock
  - Partial Failures
  - Asynchronous Communication
  - Distributed Control

Comparison of Architectures

- Traditional
  - Homogeneous Systems
  - REvolution Programming Paradigm
  - Fragile
  - Only one Transparencies
  - Closed
  - No Security
  -
  - Local Autonomy
  - Interdependence
- Distributed
  - Heterogeneous Systems
  - Evolution Programming Paradigm
  - Constant Change
  - Many Transparencies
  - Openness
  - Security
  -
  - Local Autonomy
  - Interdependence

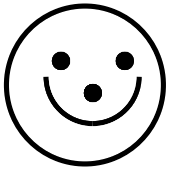
Some DADL Concepts

- *dagents* are the components
- a *contract* determines the *resources* and *performance* demands of a dagent
- *terms* and *sentences* build a *conversation* over *connections* which determines the *behavior*



S1: => (T1, T2)
S2: <= F(T1,T2)
S1;S2

Example: Plus



shared
ram

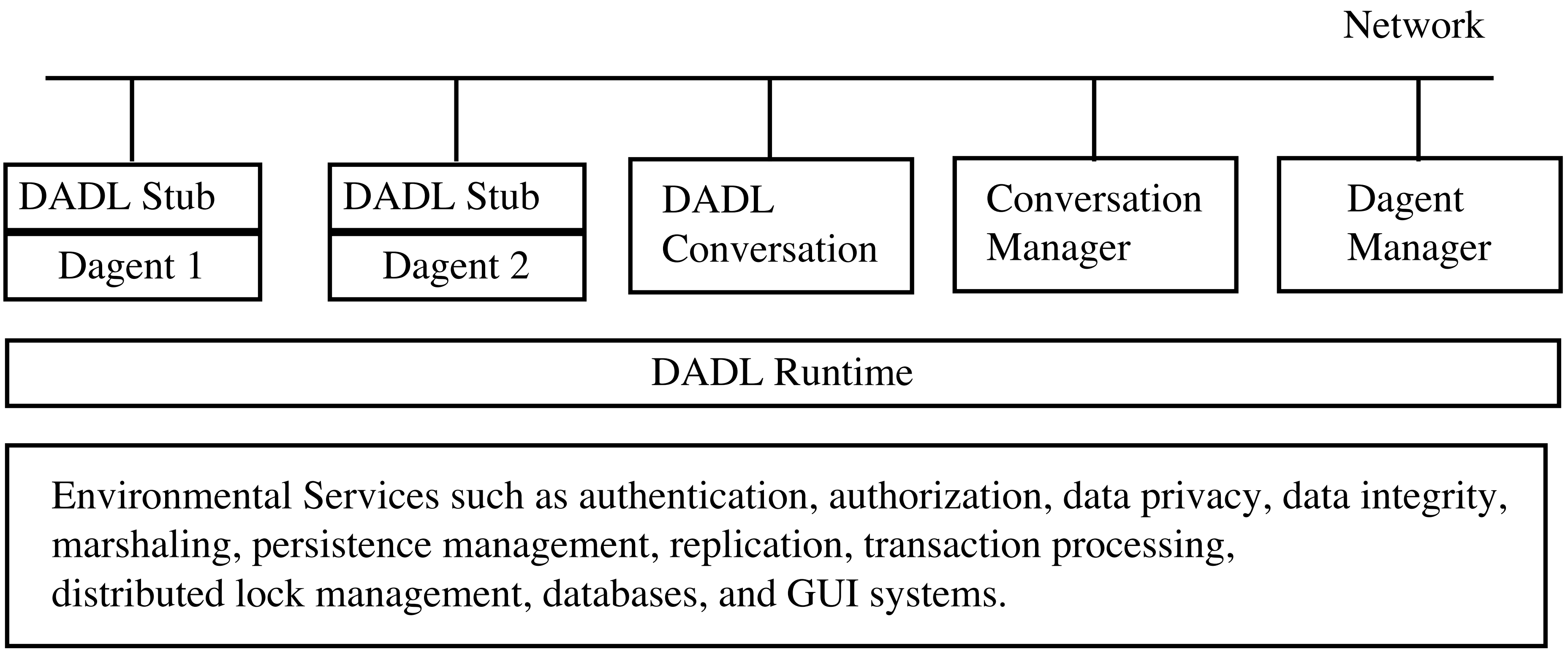
<=>
connection
c1
1-to-1
ordered FIFO
guaranteed
delivery

```
alphabet (byte);
term t1 (int n);
term t2 (int m);
sentence s1 (t1, t2) from c1 to c2;
term t3 (int plus (int, int) highly available;
sentence s2 (t3) from c2 to c1;
behavior (s1;s2);
volatile;
open data;
unmarshaled;
unauthenticated;
unauthorized;
```

<=>
connection
c2
1-to-1
ordered FIFO
guaranteed
delivery

shared
ram

A DADL Environment



Network

DADL Stub

Dagent 1

DADL Stub

Dagent 2

DADL Conversation

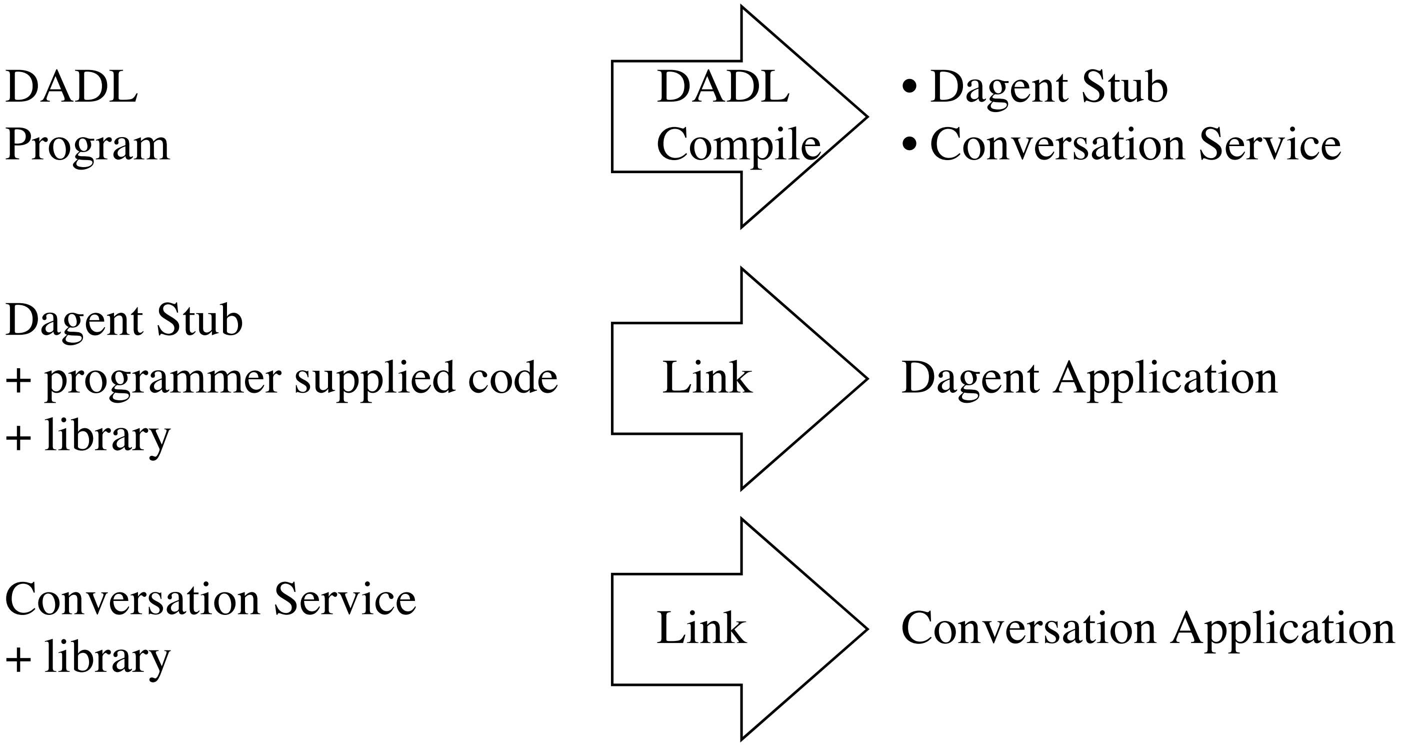
Conversation Manager

Dagent Manager

DADL Runtime

Environmental Services such as authentication, authorization, data privacy, data integrity, marshaling, persistence management, replication, transaction processing, distributed lock management, databases, and GUI systems.

DADL Development



DADL
Program

DADL
Compile

- Dagent Stub
- Conversation Service

Dagent Stub
+ programmer supplied code
+ library

Link

Dagent Application

Conversation Service
+ library

Link

Conversation Application