DeSCal–Decentarlized Shared Calendar for P2P and Ad-Hoc Networks

Jagdish Prasad Achara[Speaker],
Abdessamad Imine, Michael Rusinowitch

INRIA Nancy

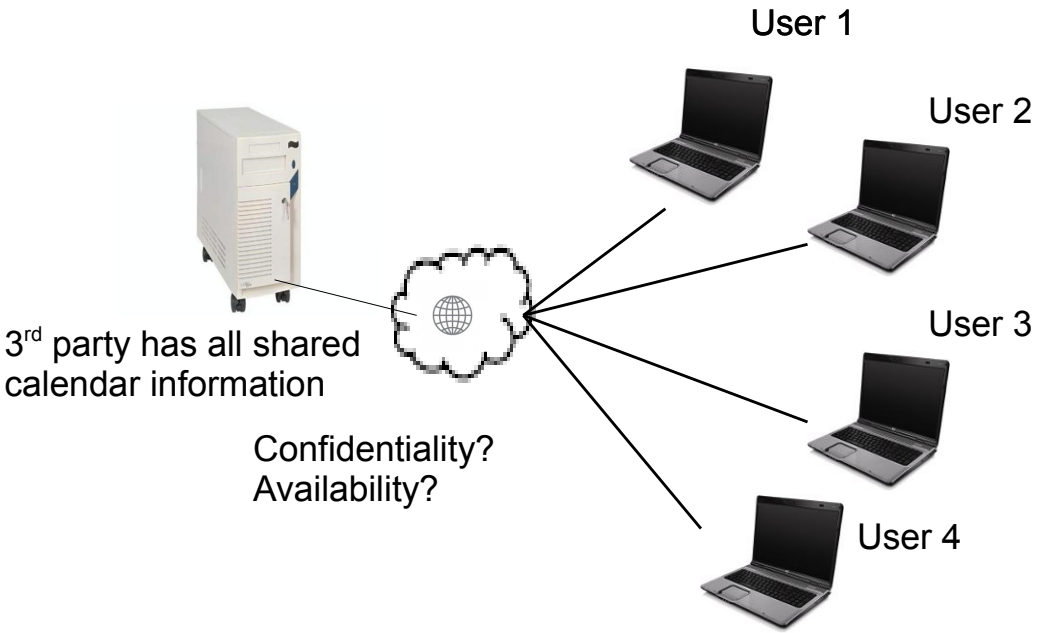
7 July, 2011

Outline

Outline

Decentralized & third party independent shared calendar

- Shared Calendar ?
- Why decentralized & third party independent ?
  1. Support for Ad-Hoc networks (802.11 networks).
  2. No single point of failure.
  3. Secrecy/confidentiality of shared calendar events.
  4. Availability of data.



Motivations & Challenges

- Motivations
  1. Design and implementation of a decentralized and third-party independent shared calendar
  2. Being a user-interactive application, the shared calendar must be-
     - Eventually Consistent and Highly Concurrent
     - Highly Responsive and Scalable

- Challenges
  1. To handle the concurrent operations on shared calendar
  2. To provide an access control mechanism on shared calendar events in a decentralized fashion

Outline

User Case Scenario and Related Work

- One real world scenario illustrating the usefulness of DeSCal (A reserach team in a research lab)

- Related Work
  - Google Calendar
  - Zimbra platform calendar application

Outline

Collaboration Model

- A user maintains two copies :
  1. the shared calendar (cooperative operations)
  2. the access control policy (administrative operations)

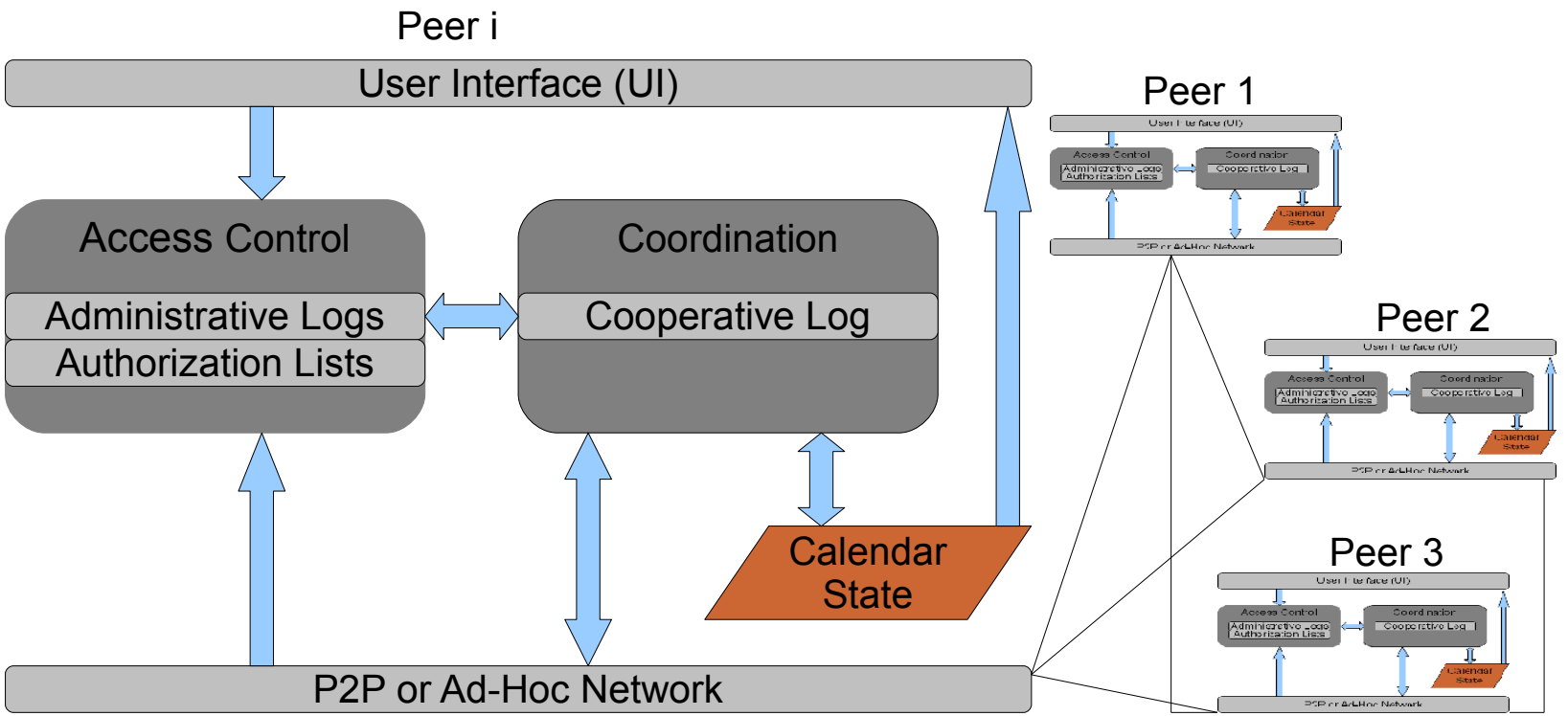
- Steps
  1. When a user changes his local copy of the shared calendar ?
  2. When a user modifies his local access control policy ?

Outline

Design

The design of DeSCal is composed of four well-seperated conceptual modules

1. Coordination Module
2. Access Control Module
3. P2P/Ad-Hoc Network Module
4. User Interface Module



Coordination Module

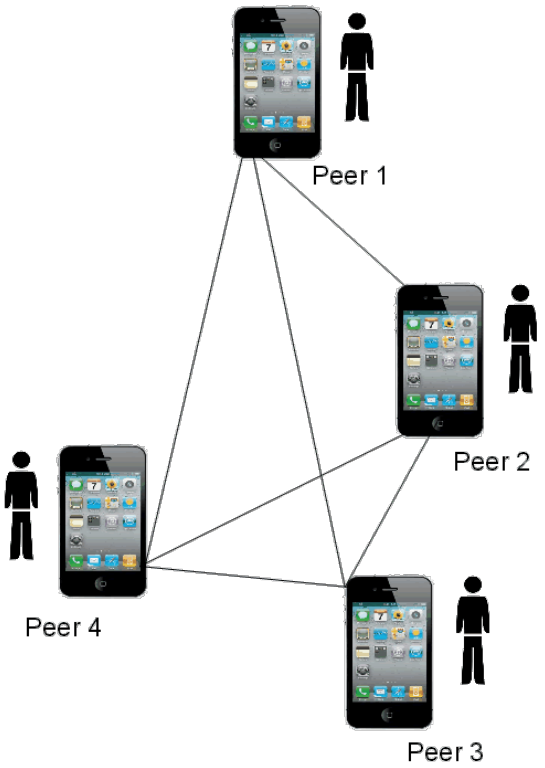
- Responsible for handling the concurrent updates on shared calendar by different users in a decentralized fashion.
- Follows a coordination model proposed by Imine for collaborative editors which is based on Operational Transformation (OT) approach.
- Directly interacts with the local copy of the shared calendar.
- Keeps track of both local and remote calendar update requests by storing them in a log called *cooperative log*.

Access Control Module

- To control access on shared calendar events so that a user is able to access the events for which he is authorized.
- Follows an access control model proposed by Imine, Cherif and Rusinowitch, however, it doesn't satisfy all the requirements of DeSCal.
- The actions taken by a user on the shared calendar through user interface has to be passed through access control module.

P2P/Ad-Hoc Network Module

- To maintain a local knowledge of the network infrastructure.
- Responsible for providing Peer-to-Peer distributed architecture services to DeSCal for any kind of network.

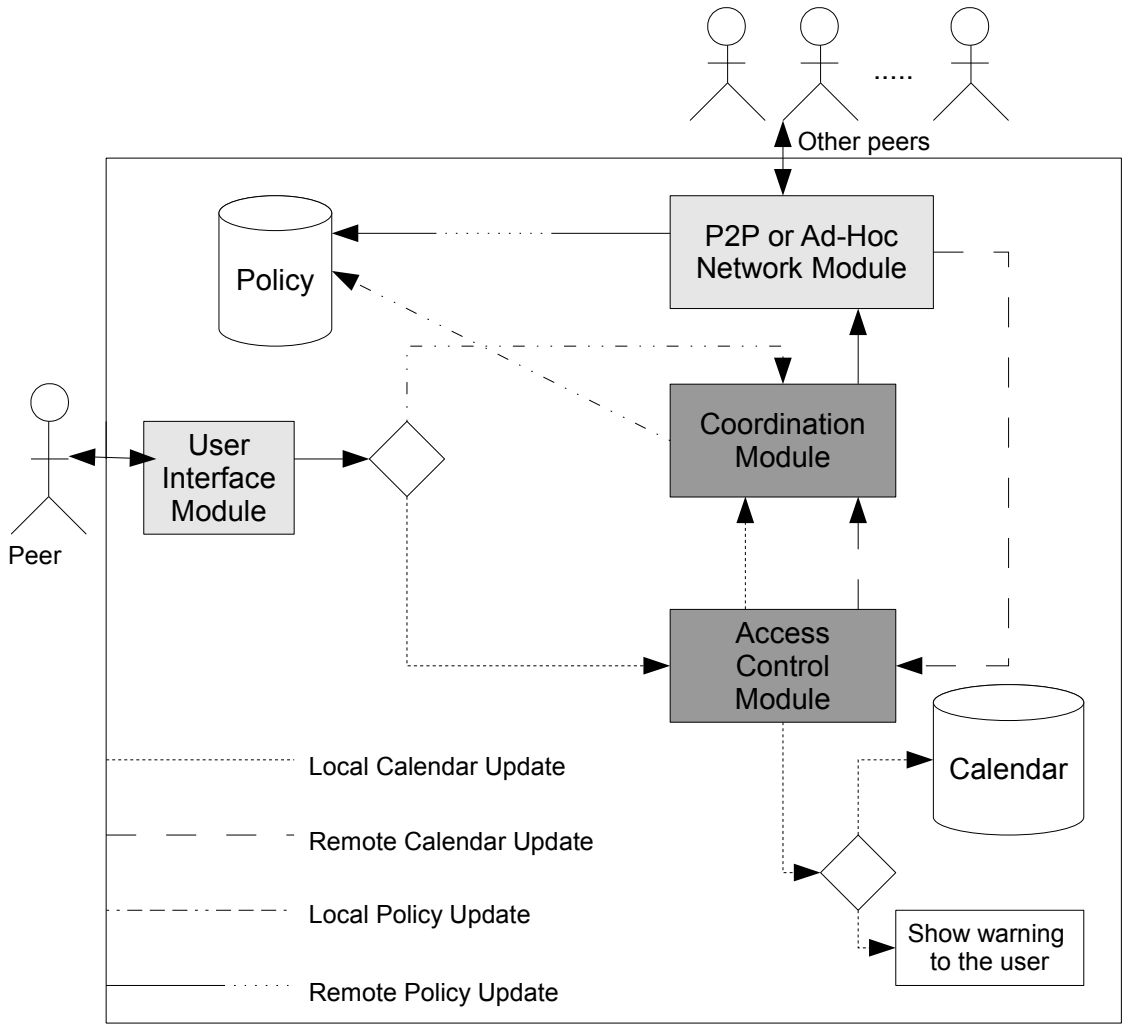


User-Interface Module

- Enables a user to take actions on the shared calendar. However, it can't change the state of the shared calendar directly.
- Should be designed in such a way that simplifies the use of DeSCal for the user.

Architecture

How does a user handle the local updates generated by himself and remote updates by other users in the group received through P2P/Ad-Hoc Network module ?



Outline

Implementation on iPhone OS

- Key entities
  1. Calendar
  2. Event
  3. Rule
  4. Policy

- Working
  1. Inserting a new event
  2. Managing Policy
  3. Deleting/Editing an event

Implementation on iPhone OS

FIGURE: Calendar, Event Detail, Policy and Available Peers view

Implementation on iPhone OS

FIGURE: Selection of various attributes to insert a new rule in iPhone OS implementation

Outline

Various Scenarios demonstrating application's novelty

- No constrain on number of users & dynamic access control
- Eventually, the consistency of replicated copies of shared calendar is achieved in all cases.
- Users leave or join the group independently.

Outline

Future Work

- Future Work
  - Providing confidentiality to replicated shared calendar events.
  - Securing the communication between users.