**Problems**

- Developers manually craft the data management logic into each app!
- App logic and data management logic are intertwined.
- Users don’t have control over their data.

**BlueMountain Vision**

- Two kinds of user installable apps
- Regular app implements only core-app logic
- **Data management app** manages the data and data policies
- Users can link any data management app with any regular app!

*Easy to develop apps; users have more control; easy to deploy new data management solutions!*

**Users can link any data management app with any regular app!**

**Workflow**

- BlueMountain transformer, based on Reptor [1], can transform any app into a ‘BlueMountain enabled app’ by injecting our framework into the app
- ‘BlueMountain enabled apps’ can be linked with any data management app by the users based on their needs.

**Architecture**

- **BlueMountain API**: All data management apps implement this
  - Files: Object based API, suitable for cloud
  - Database: SQL statement based API
  - Key/Value pair: Developed on top of files
- **Translation Layer**: Translates Android APIs to BlueMountain APIs
- **Dynamic Loading**: Data management modules are loading into an app at runtime and works in the same address space.
- **Runtime policies**: Can be set by IPC (Android Binder framework)

**Use Cases**

- Custom data management apps based on previous work can be implemented.
- We have implemented five data management apps for BlueMountain
  - Database sync systems based on Operation logs, File sync & Multiple-databases.

**Evaluation**

- **Use Case Testing**
  - Photo Talks app, a popular photo editing app on Android, is transformed to a BlueMountain enabled app.
  - Can be linked with any of our test data management apps.
  - Photos are redirected from local storage to desktop and Dropbox!

- **Tests and instrumentation**
  - Tested with 400 apps downloaded from Google Play.
  - All apps were instrumented by injecting BlueMountain framework.
  - Overhead due to translation, framework and dynamic loading indicate a reasonable overhead of 10-30%.

**References**