API Automation

Alex Bowen
Community & Developer Relations
dreamfactory.com

dreamfactory

Lessons learned

API automation

Examples

Lessons learned

Typical enterprise development project

More than HALF of time spent on backend integration, API development, and testing!

Creating APIs manually for each new project

Slows you down

The problem is compounded by mobile projects

Big enterprises need to deploy THOUSANDS of mobile apps!

Typical workflow of manually coding APIs….

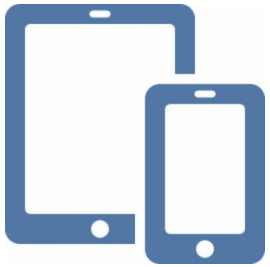
App 1

SQL Database

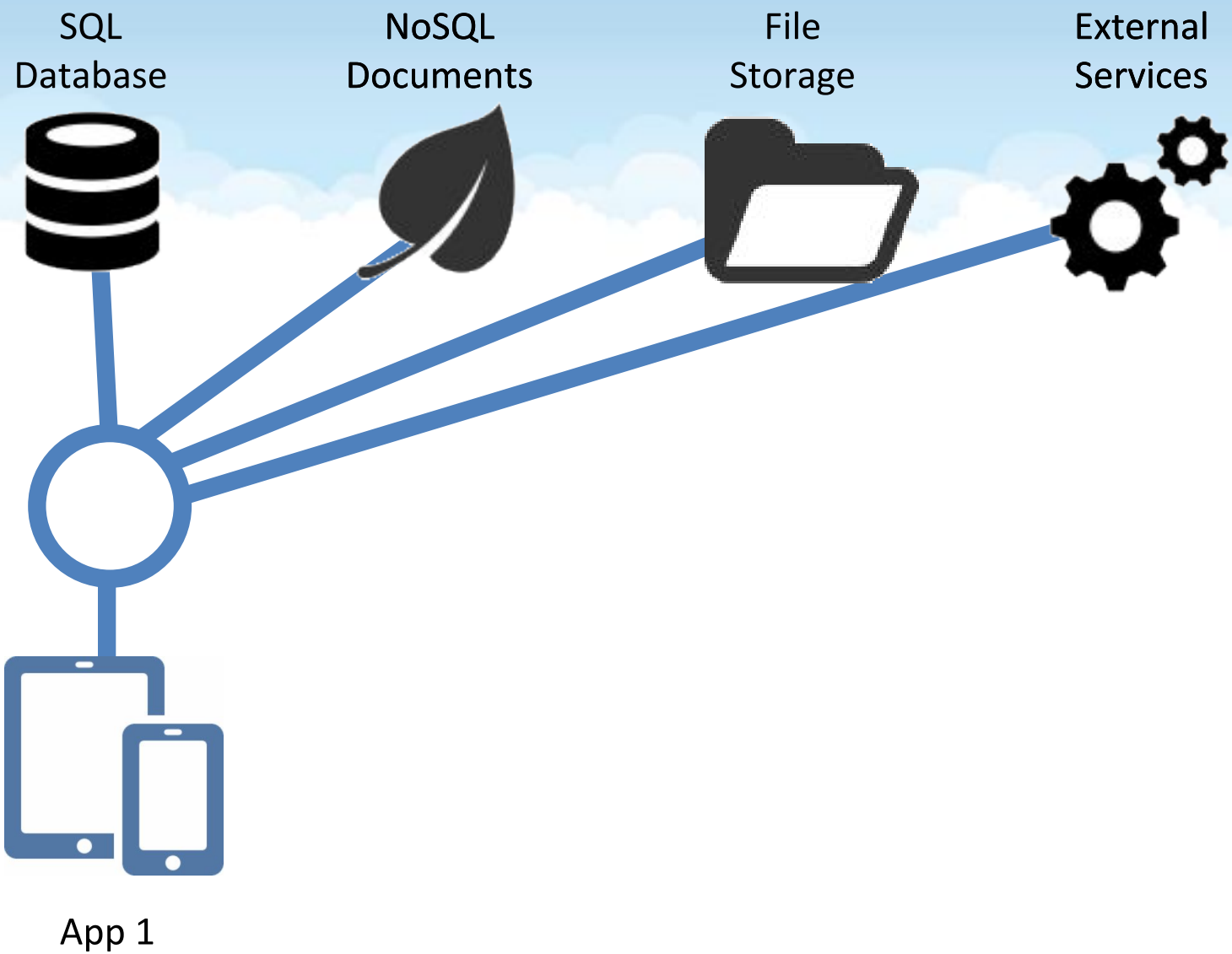
NoSQL Documents

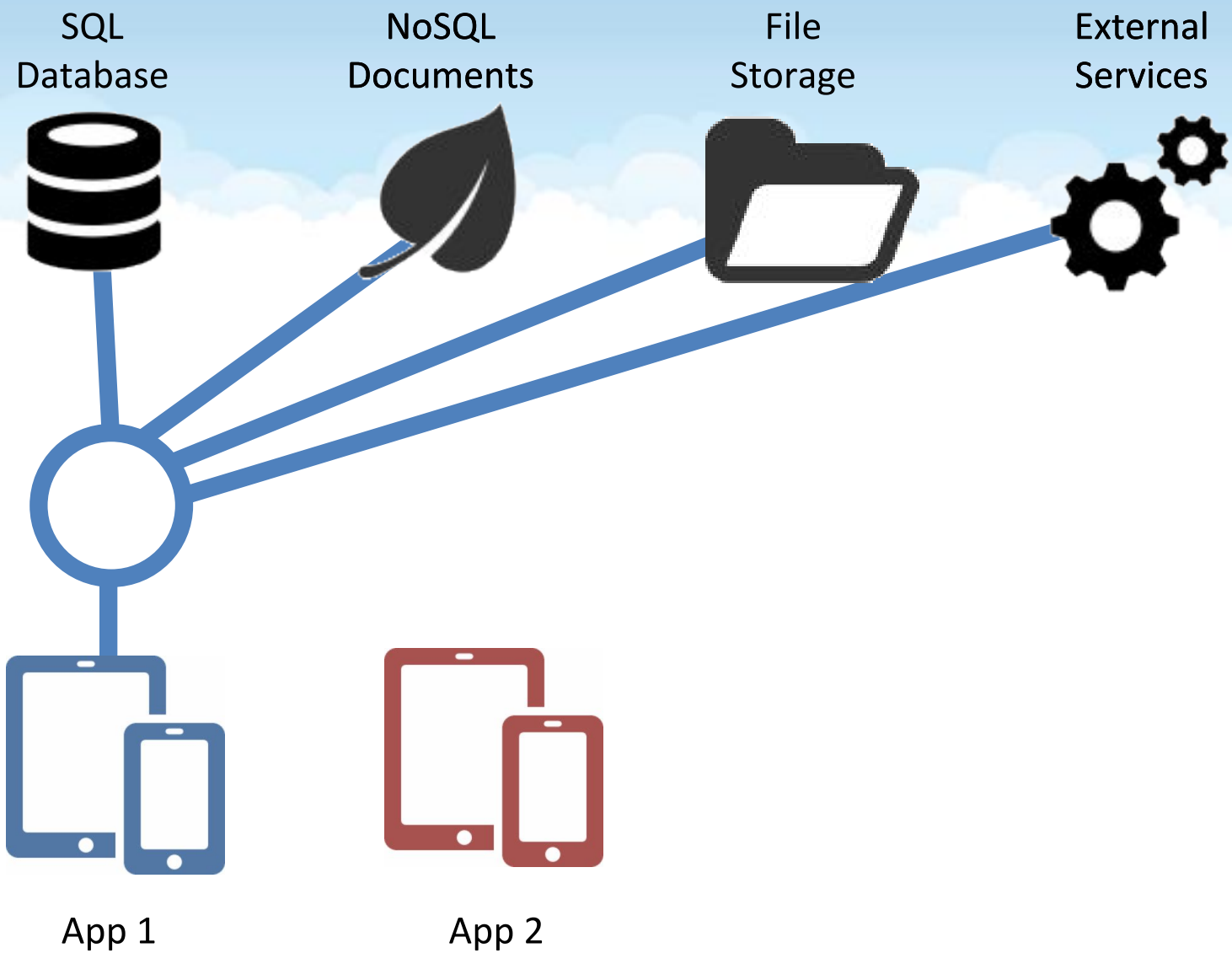
File Storage

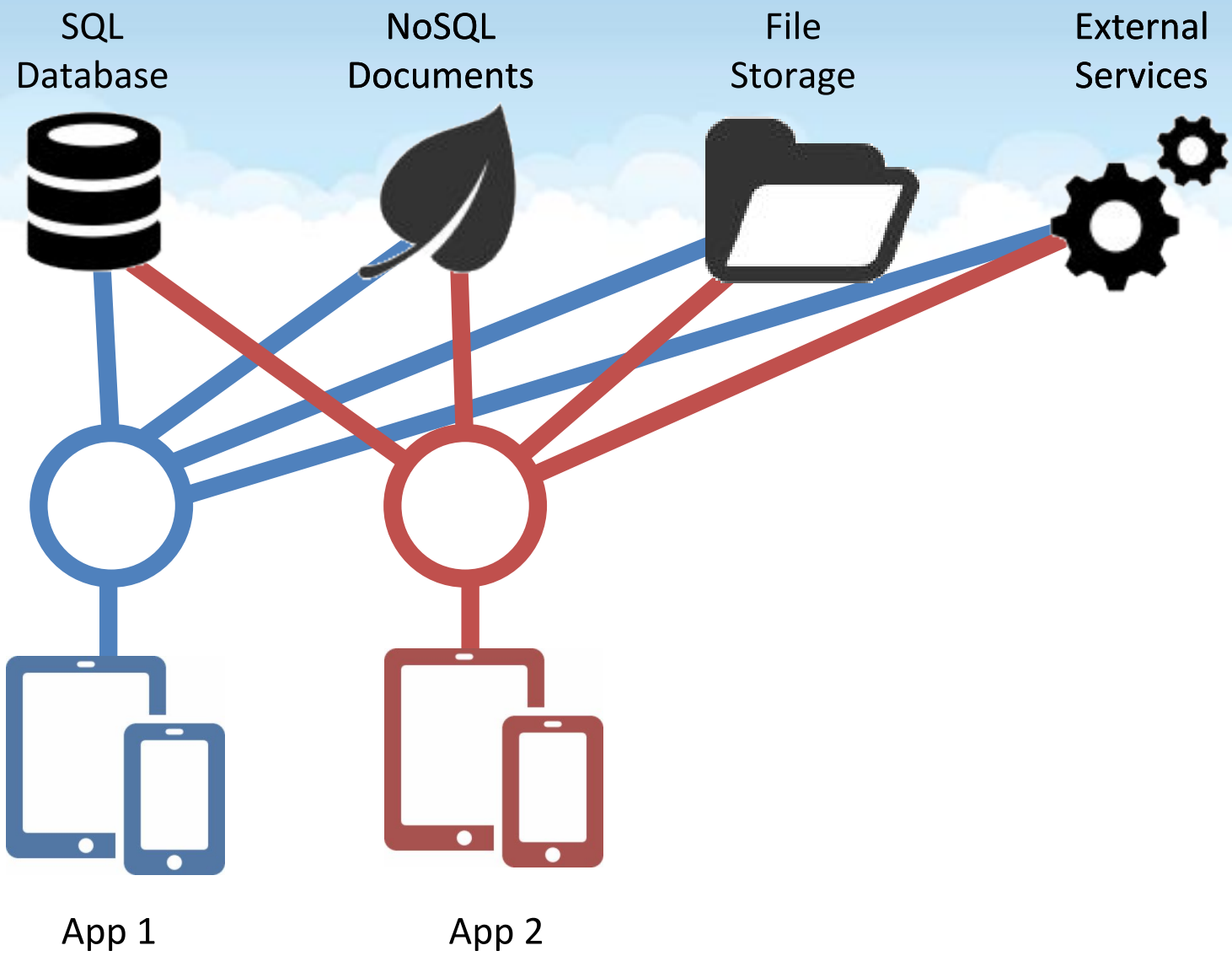
External Services

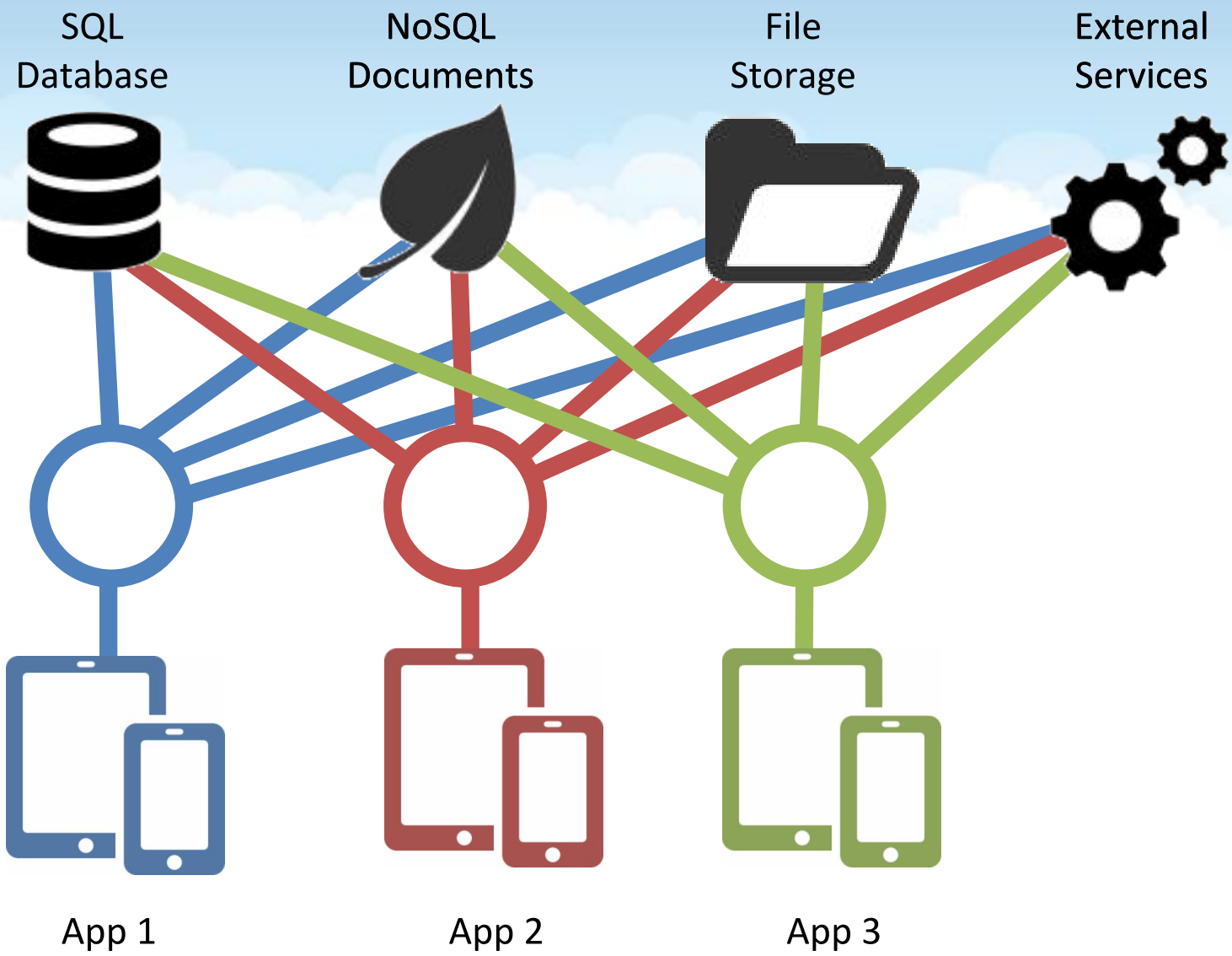


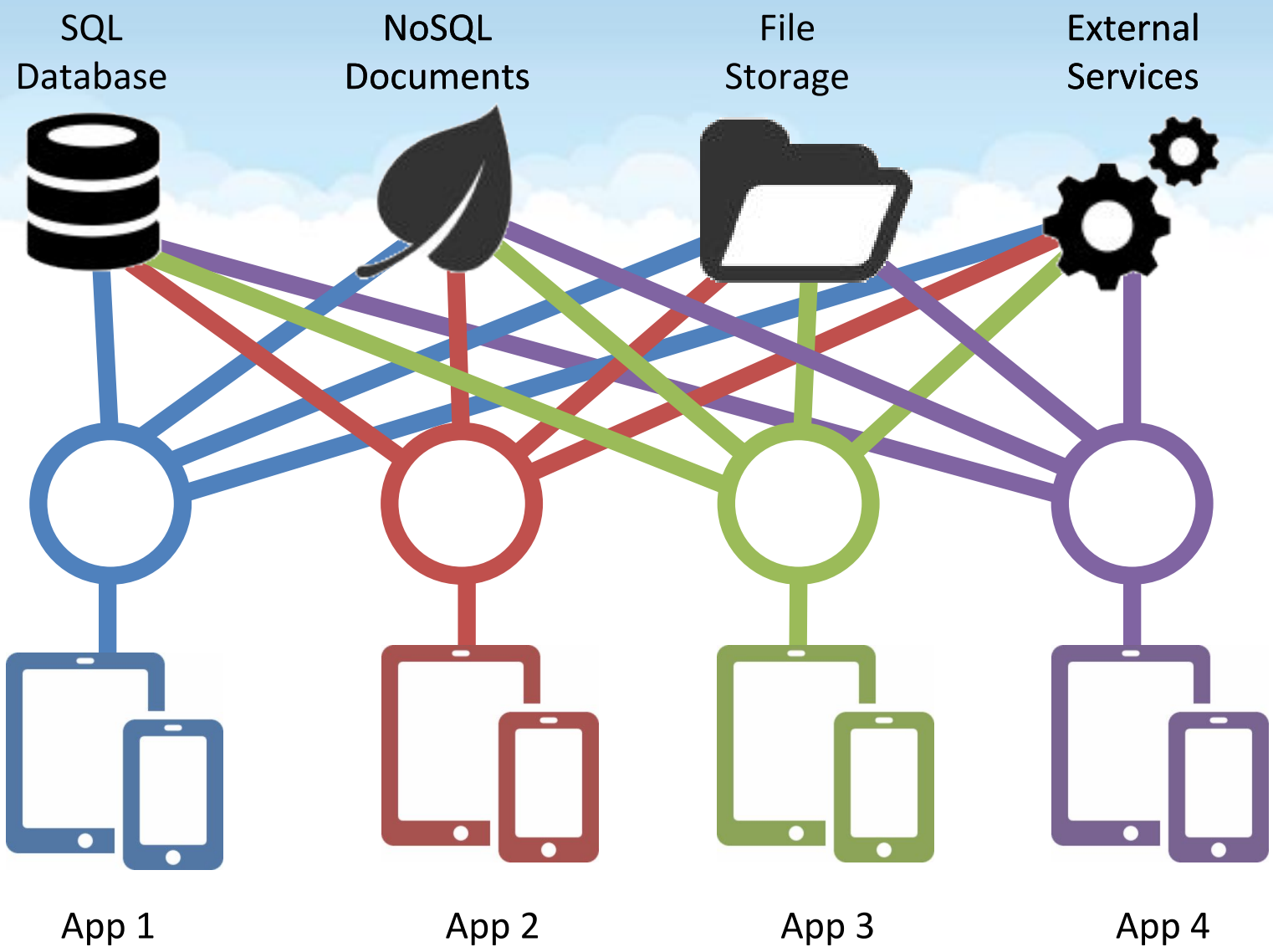
App 1


SQL
Database
NoSQL
Documents
File
Storage
External
Services
App 1

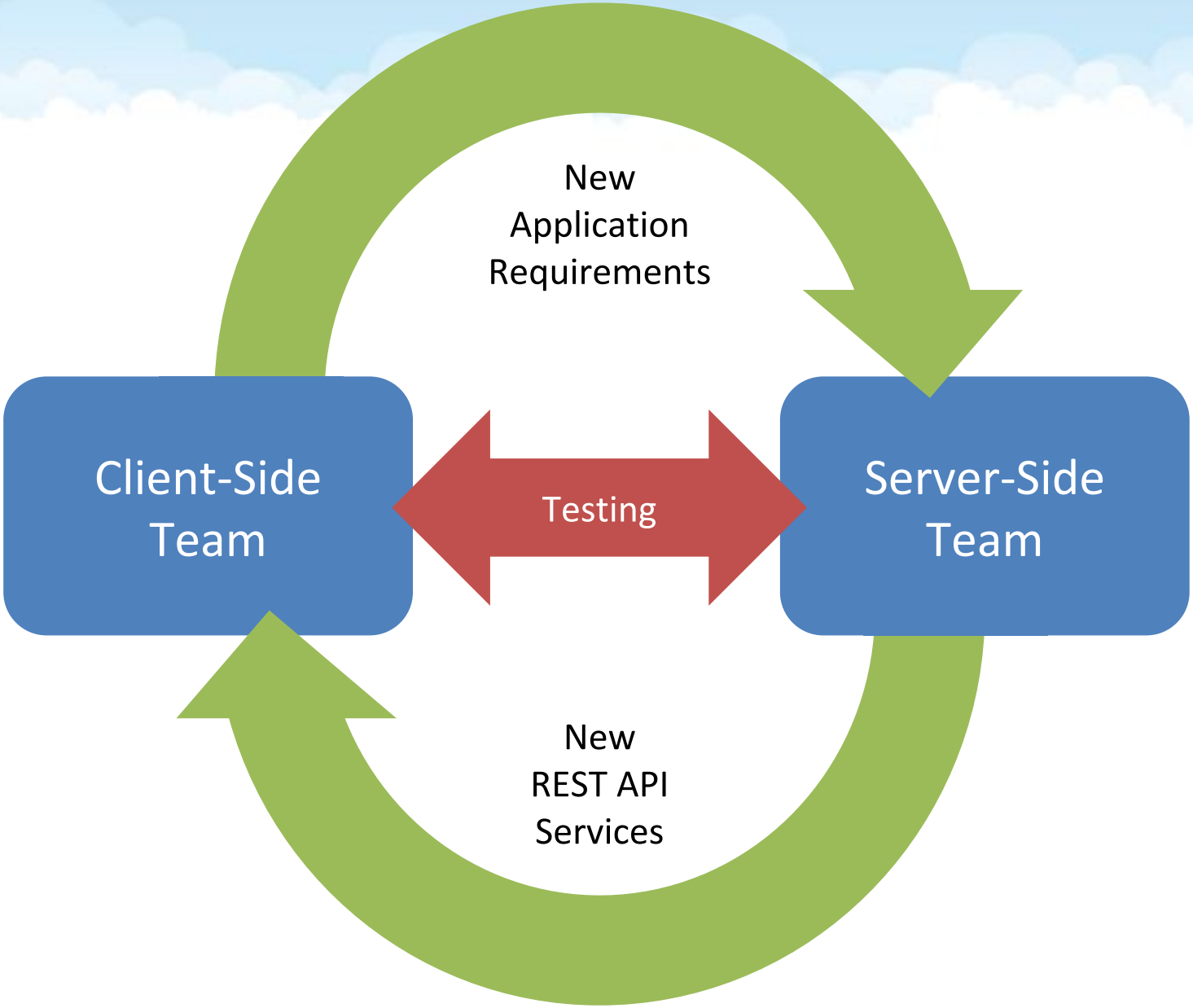

SQL
Database
NoSQL
Documents
File
Storage
External
Services
App 1
App 2


SQL
Database
NoSQL
Documents
File
Storage
External
Services
App 1
App 2


SQL
Database
NoSQL
Documents
File
Storage
External
Services
App 1
App 2
App 3


SQL
Database
NoSQL
Documents
File
Storage
External
Services
App 1
App 2
App 3
App 4

The Interface Negotiation



Manual API Complexity

New APIs for every project
Tightly linked to data sources
Tightly linked to backend infrastructure
Poorly documented
Difficult to scale, not portable
Many security vulnerabilities
Server-side software development
Time consuming interface negotiation

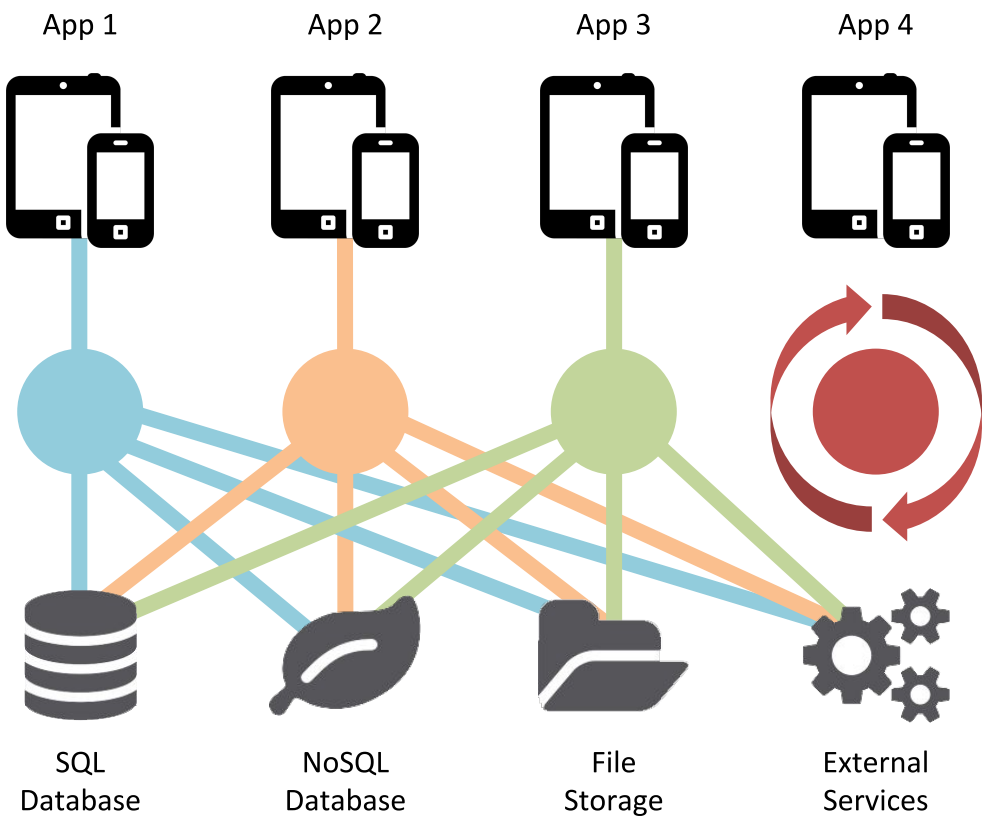
Complexity Band-Aids

- API Creation Tools
  - Generates more complexity faster
- API Management Tools
  - Introduces additional proxy endpoint
  - Increases overall complexity of the system
  - Still writing one-off APIs!
- Mobile Device Management
  - Control the data, not the devices

To Recap...

Problems manually coding APIs

- Companies waste time and money on backend integration
- Increasing backend complexity reduces security, reliability and portability
- Interface negotiation between client-side and server-side teams causes friction



Building new REST APIs for every new project is a bad idea

What is API automation?

API automation to the rescue

Reusable APIs for each new project
speeds you up

Invert the problem

Start with the data, not the apps.

SQL Database

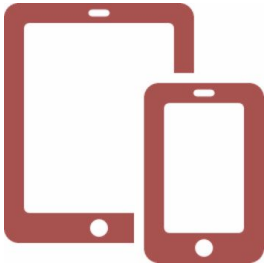
NoSQL Documents

File Storage

External Services

Automated Set of Reusable REST APIs

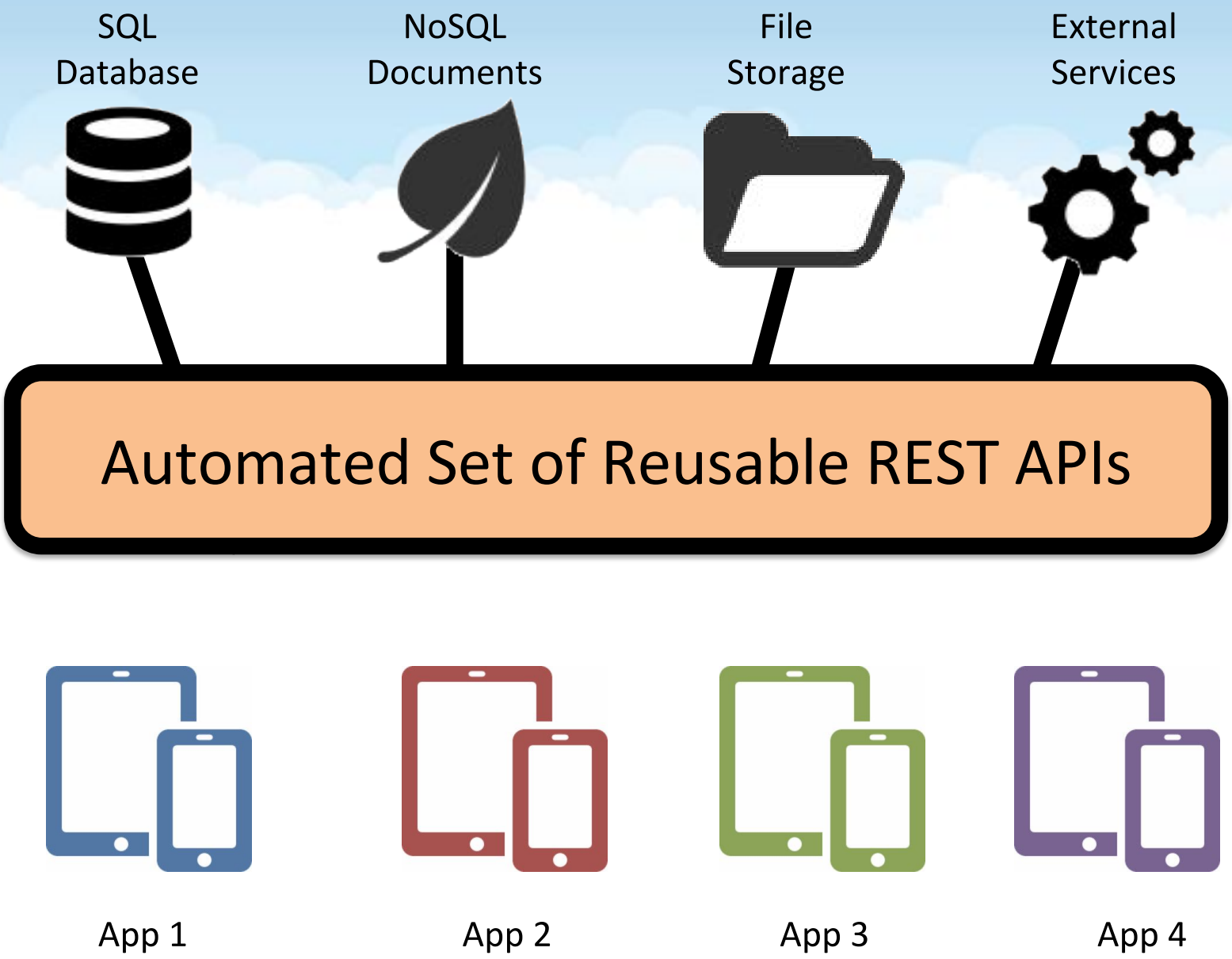
App 1

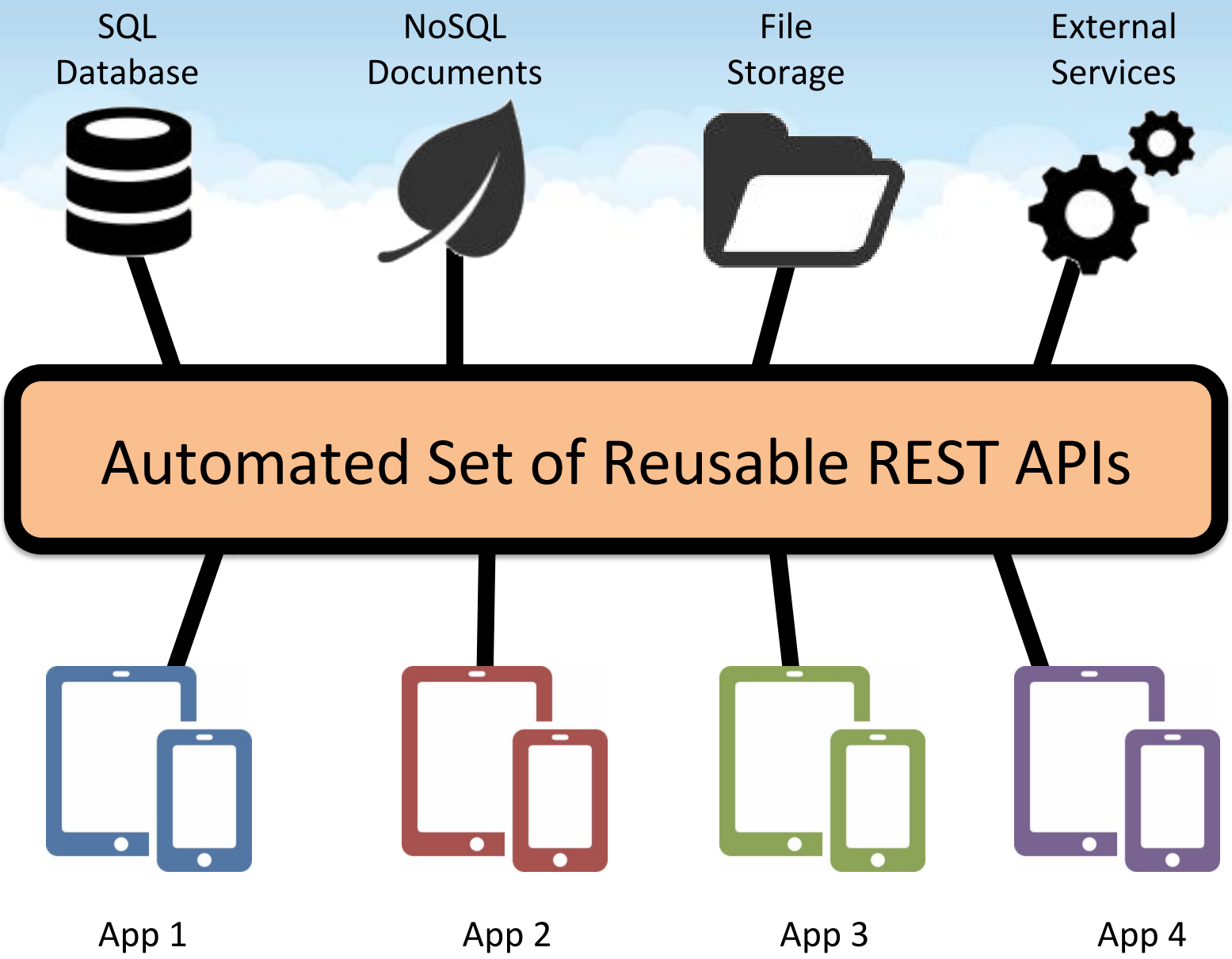


App 2

App 3

App 4


SQL Database
NoSQL Documents
File Storage
External Services
Automated Set of Reusable REST APIs
App 1
App 2
App 3
App 4


SQL Database
NoSQL Documents
File Storage
External Services
Automated Set of Reusable REST APIs
App 1
App 2
App 3
App 4

API Automation Approach

Reusable APIs for any new project

Customization for special cases

Focus on front-end app development

Decouple client-side from server-side

Flexible backend infrastructure

Flexible backend data sources

Automatically documented

Scalable, reliable, portable, secure

What is an automated REST API?

Automated REST APIs

Request URL

Server Version Database Table

```
https://www.myserver.com/api/v2/oracleDB/Contacts?filter=lastName%20like%20'jon%'&order=firstName
```

Filter Parameter Other Parameters

Number of APIs generated

- SQL 45
- NoSQL 35
- File storage 15
- Any number of external services
- User management 15
- System management 80

Deep support for legacy SQL

- Rollback and commit
- Pagination and ordering
- Complex filters
- Metadata
- Stored procedures
- Related objects
- Virtual foreign keys

Automatically created for SQL, NoSQL, files, cache, email, push notifications

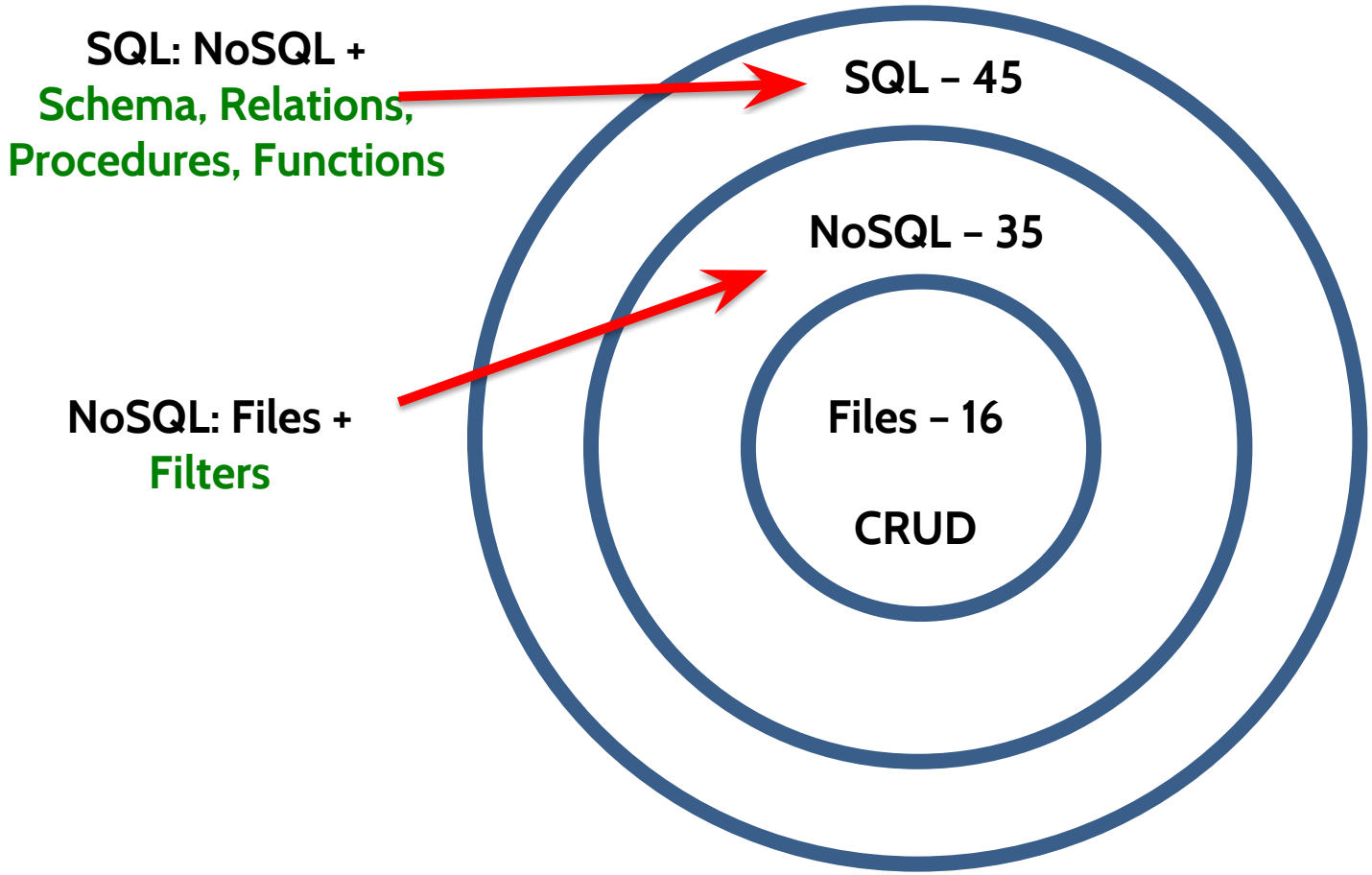
Simple AND flexible endpoints

Decouples back-end from front-end

Consistently structured for SQL, NoSQL, file stores

Supports any client technology: web, mobile, IoT

Manageable number of API endpoints

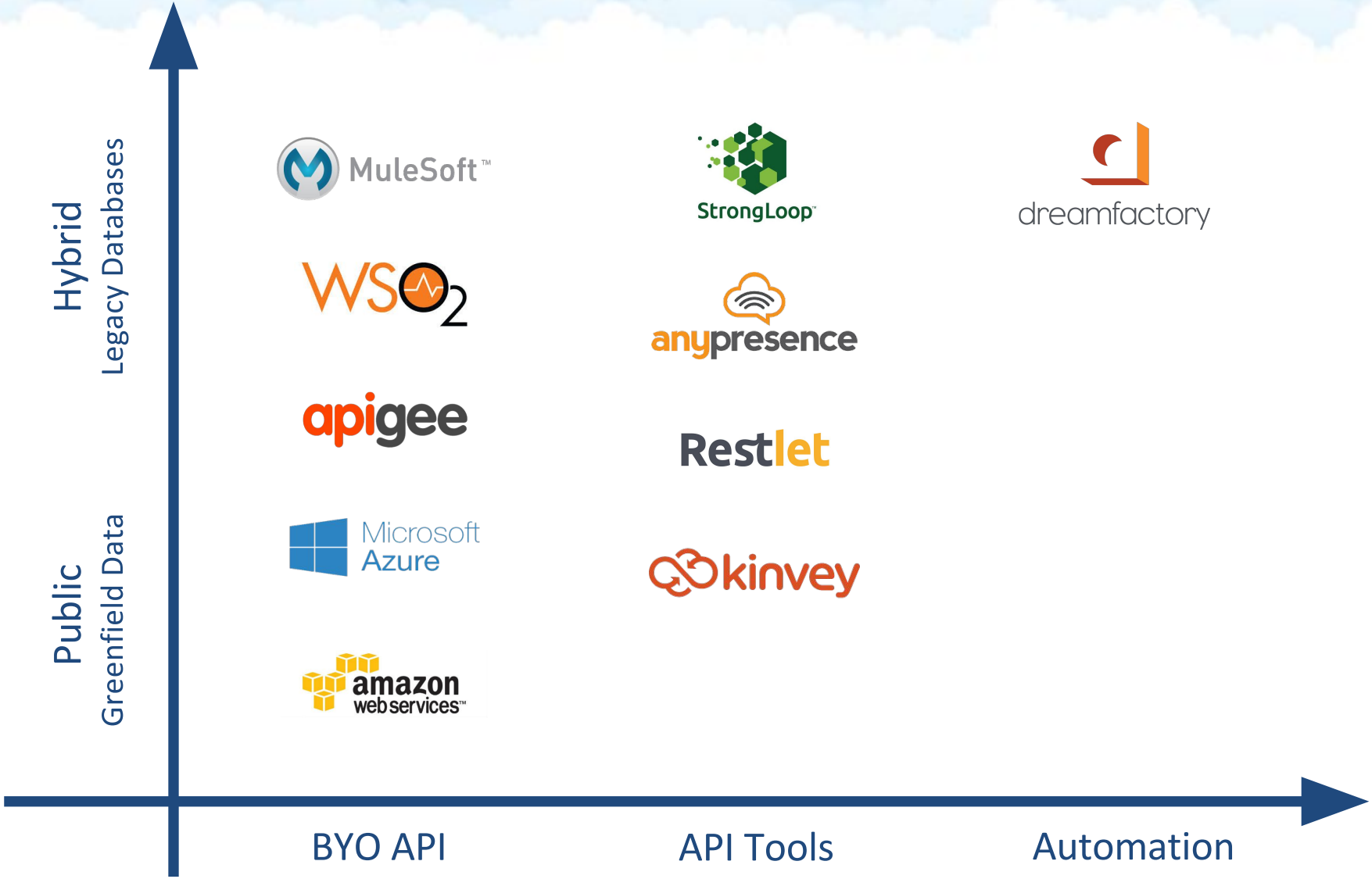


Append parameters to endpoints

Dates
Filters
Pagination
Relationships
Files
Advanced Use Cases

Handle business logic with server-side scripts on API endpoints

API Landscape

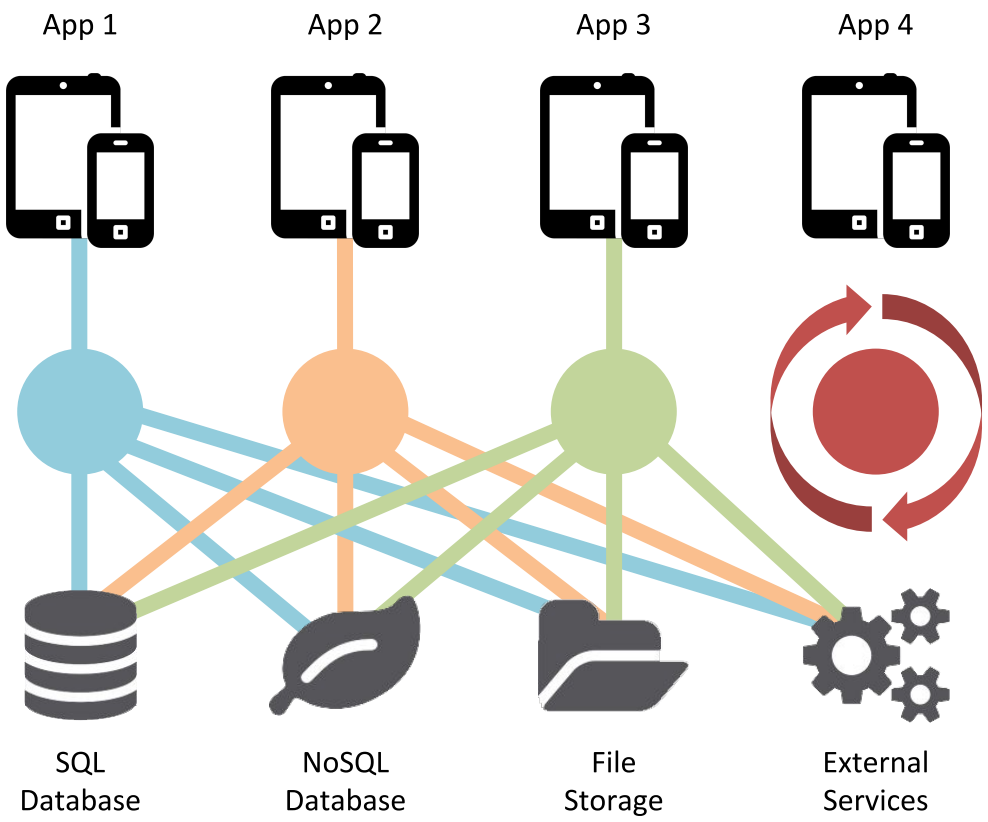


Demo time!

To Recap...

Problems manually coding APIs

- Companies waste time and money on backend integration
- Increasing backend complexity reduces security, reliability and portability
- Interface negotiation between client-side and server-side teams causes friction



Building new REST APIs for every new project is a bad idea

If you remember one thing....

Simplify development with API automation!

Thank You!

alexbowen@dreamfactory.com
@AlexandraABowen

SQL API Examples

Multiple records

GET	/db/{table_name}	getRecordsByFilter() - Retrieve one or more records by using a filter.
GET	/db/{table_name}	getRecordsByIds() - Retrieve one or more records by identifiers.
POST	/db/{table_name}	getRecordsByPost() - Retrieve one or more records by posting necessary data.
GET	/db/{table_name}	getRecords() - Retrieve one or more records.
POST	/db/{table_name}	createRecords() - Create one or more records.
PUT	/db/{table_name}	replaceRecordsByIds() - Update (replace) one or more records.
PUT	/db/{table_name}	replaceRecordsByFilter() - Update (replace) one or more records.
PUT	/db/{table_name}	replaceRecords() - Update (replace) one or more records.

Single records

GET	/db/{table_name}/{id}	getRecord() - Retrieve one record by identifier.
POST	/db/{table_name}/{id}	createRecord() - Create one record with given identifier.
PUT	/db/{table_name}/{id}	replaceRecord() - Replace the content of one record by identifier.
PATCH	/db/{table_name}/{id}	updateRecord() - Update (patch) one record by identifier.
DELETE	/db/{table_name}/{id}	deleteRecord() - Delete one record by identifier.

Stored procedures & functions

GET	/db/_proc	getStoredProcs() - List callable stored procedures.
GET	/db/_proc/{procedure_name}	callStoredProc() - Call a stored procedure.
POST	/db/_proc/{procedure_name}	callStoredProcWithParams() - Call a stored procedure.
GET	/db/_func	getStoredFuncs() - List callable stored functions.
GET	/db/_func/{function_name}	callStoredFunc() - Call a stored function.
POST	/db/_func/{function_name}	callStoredFuncWithParams() - Call a stored function.

NoSQL API Examples

Multiple documents

GET	/dynamodb/{table_name}	getRecordsByFilter() - Retrieve one or more records by using a filter.
GET	/dynamodb/{table_name}	getRecordsByIds() - Retrieve one or more records by identifiers.
POST	/dynamodb/{table_name}	getRecordsByPost() - Retrieve one or more records by posting necessary data.
GET	/dynamodb/{table_name}	getRecords() - Retrieve one or more records.
POST	/dynamodb/{table_name}	createRecords() - Create one or more records.
PUT	/dynamodb/{table_name}	replaceRecordsByIds() - Update (replace) one or more records.
PUT	/dynamodb/{table_name}	replaceRecordsByFilter() - Update (replace) one or more records.
PUT	/dynamodb/{table_name}	replaceRecords() - Update (replace) one or more records.

Single documents

GET	/dynamodb/{table_name}/{id}	getRecord() - Retrieve one record by identifier.
POST	/dynamodb/{table_name}/{id}	createRecord() - Create one record with given identifier.
PUT	/dynamodb/{table_name}/{id}	replaceRecord() - Replace the content of one record by identifier.
PATCH	/dynamodb/{table_name}/{id}	updateRecord() - Update (patch) one record by identifier.
DELETE	/dynamodb/{table_name}/{id}	deleteRecord() - Delete one record by identifier.

File API Examples

Multiple files

GET	/S3/{container}/{folder_path}/	getFolder() - List the folder's content, including properties.
POST	/S3/{container}/{folder_path}/	createFolder() - Create a folder and/or add content.
PATCH	/S3/{container}/{folder_path}/	updateFolderProperties() - Update folder properties.
DELETE	/S3/{container}/{folder_path}/	deleteFolder() - Delete one folder and/or its contents.

Single files

GET	/S3/{container}/{file_path}	getFile() - Download the file contents and/or its properties.
POST	/S3/{container}/{file_path}	createFile() - Create a new file.
PUT	/S3/{container}/{file_path}	replaceFile() - Update content of the file.
PATCH	/S3/{container}/{file_path}	updateFileProperties() - Update properties of the file.
DELETE	/S3/{container}/{file_path}	deleteFile() - Delete one file.