# Building Controls Applications Using HTTP Services
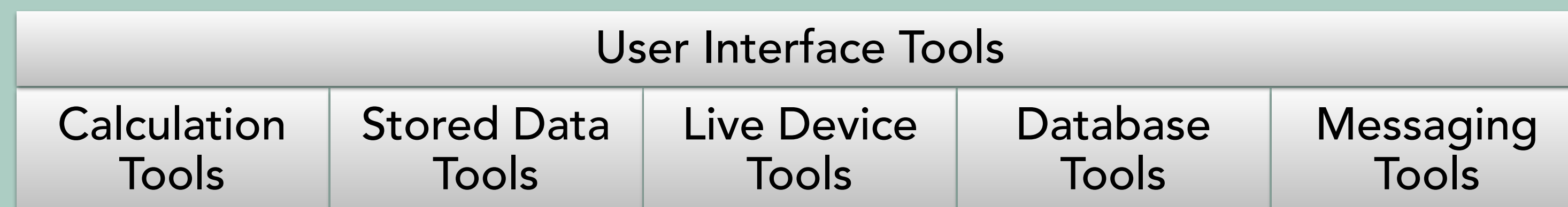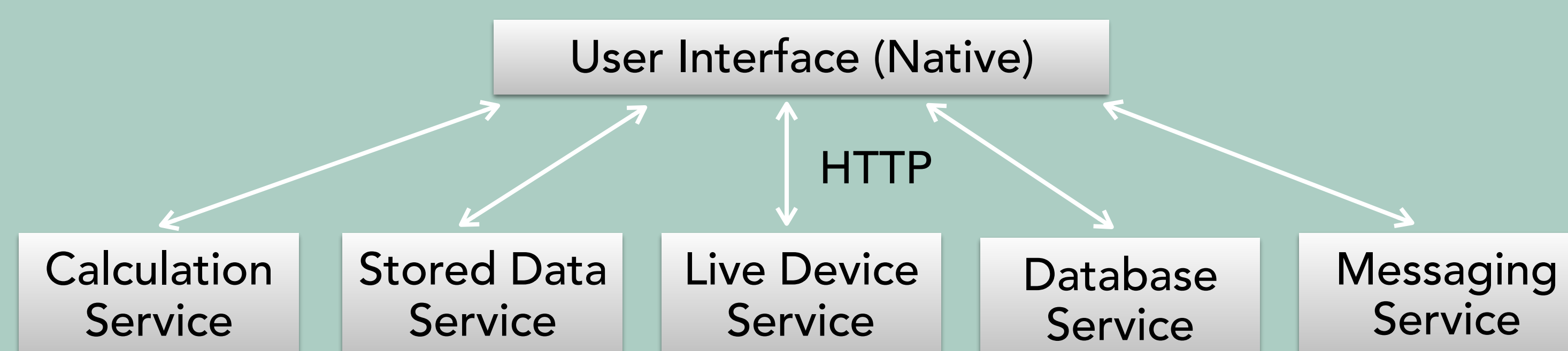
## T. D'Ottavio, K. Brown, A. Fernando, S. Nemesure

**Traditional Monolithic Application**

User Interface Tools

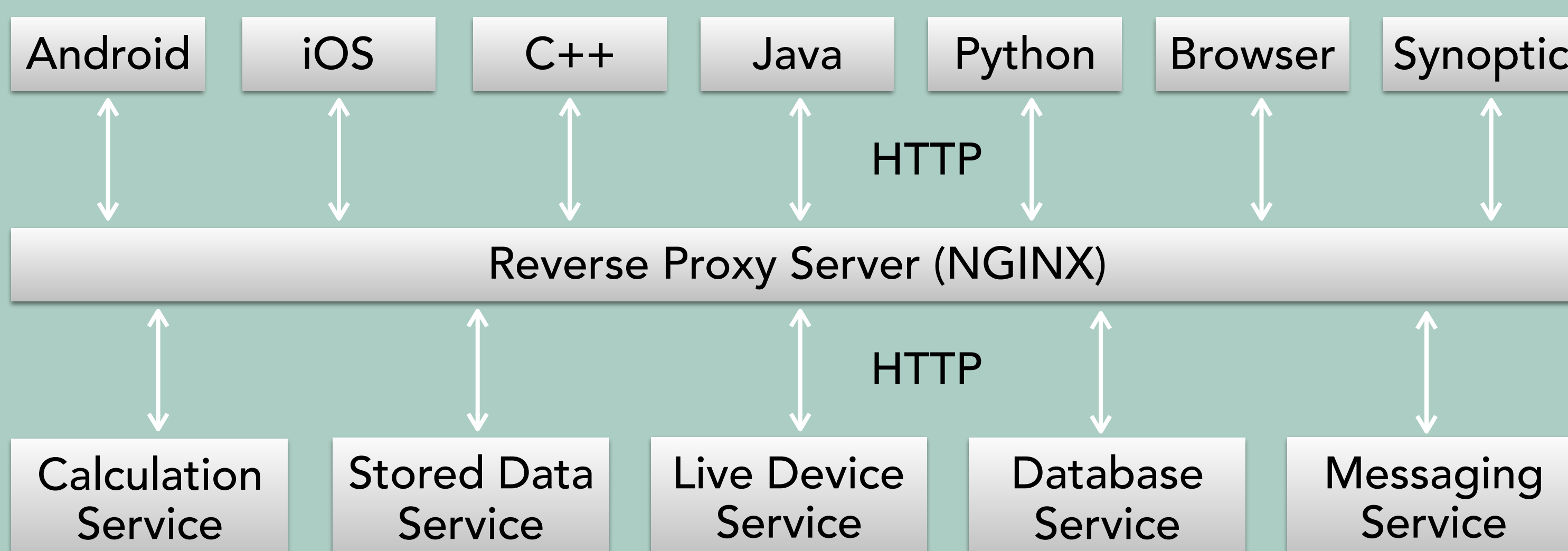| Calculation Tools | Stored Data Tools | Live Device Tools | Database Tools | Messaging Tools |
| --- | --- | --- | --- | --- |

- Application and services (tools) are written in the same language (usually C++ or Java) and bound into a single executable.
- Advantages - performance, troubleshooting

**Application Built with Networked HTTP Services**

User Interface (Native)

HTTP

| Calculation Service | Stored Data Service | Live Device Service | Database Service | Messaging Service |
| --- | --- | --- | --- | --- |

- Client applications talk to services using HTTP and a RESTful interface.
- Services are tools written in any language and wrapped with an HTTP service shell.
- Advantages
  - supports a wide variety of clients
  - can reuse toolkits as networked services
  - easy to assign service responsibility

**Multiple Applications Sharing HTTP Services**

| Android | iOS | C++ | Java | Python | Browser | Synoptic |
| --- | --- | --- | --- | --- | --- | --- |

HTTP

Reverse Proxy Server (NGINX)

HTTP

| Calculation Service | Stored Data Service | Live Device Service | Database Service | Messaging Service |
| --- | --- | --- | --- | --- |

- A Reverse Proxy Server allows networked services to be shared by many applications.
- The server:
  - routes clients to the appropriate service
  - permits remote and mobile clients
  - provides load balancing where needed
  - limits connections to prevent DOS attacks
  - provides added security, SSL termination

BROOKHAVEN
NATIONAL LABORATORY