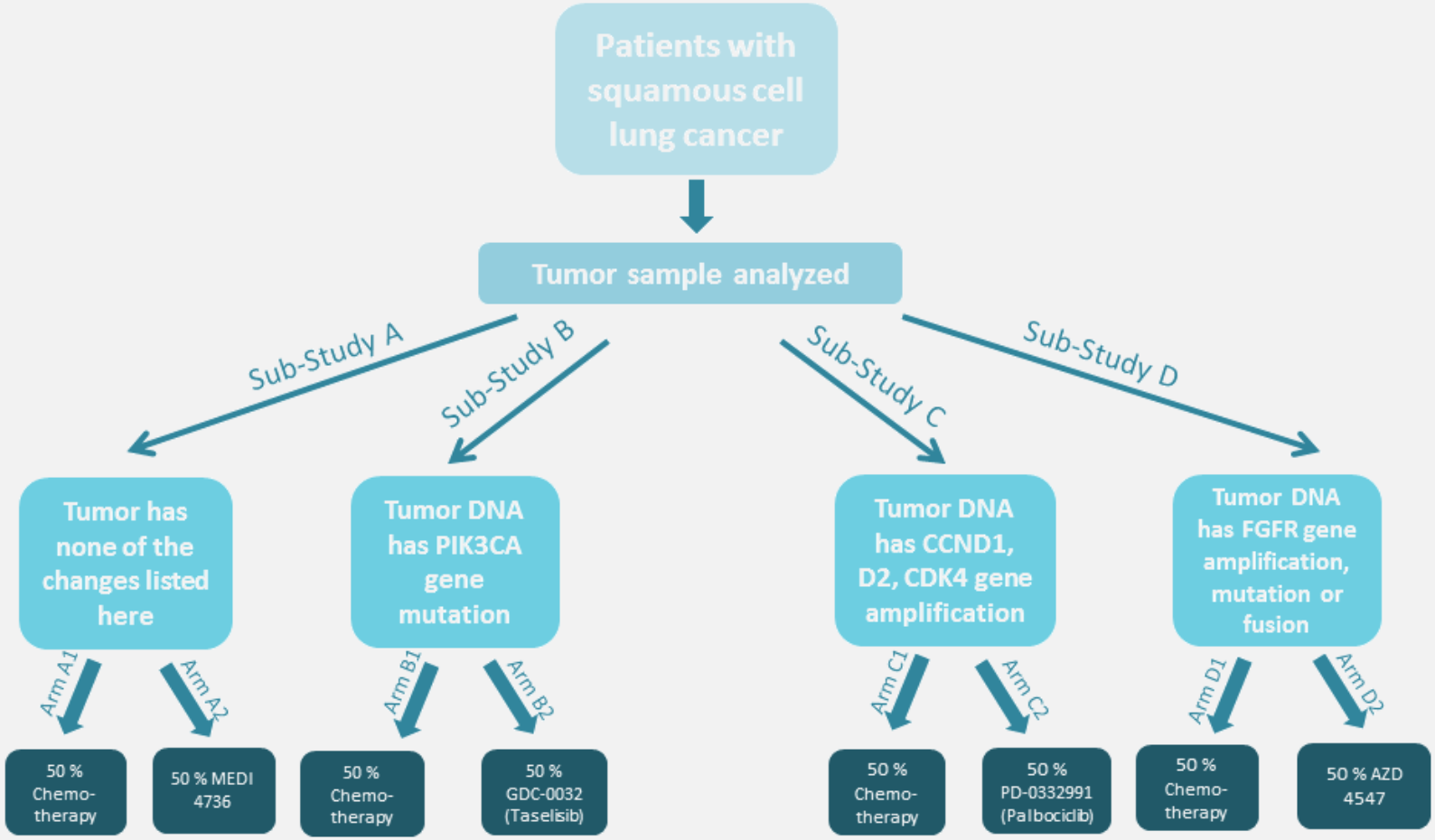
TECHNICAL IMPLEMENTATION OF LUNG-MAP

Chris Cook, Taylor Phillips, Angela Smith, Yoko Rivers, Dave Law

STUDY DESIGN

- Lung-MAP study design is atypical
- A biomarker analysis and sub-study assignment needs to occur before treatment assignment
- Involves multiple trials, not just 1

Lung-MAP Sub-Studies for Treatment

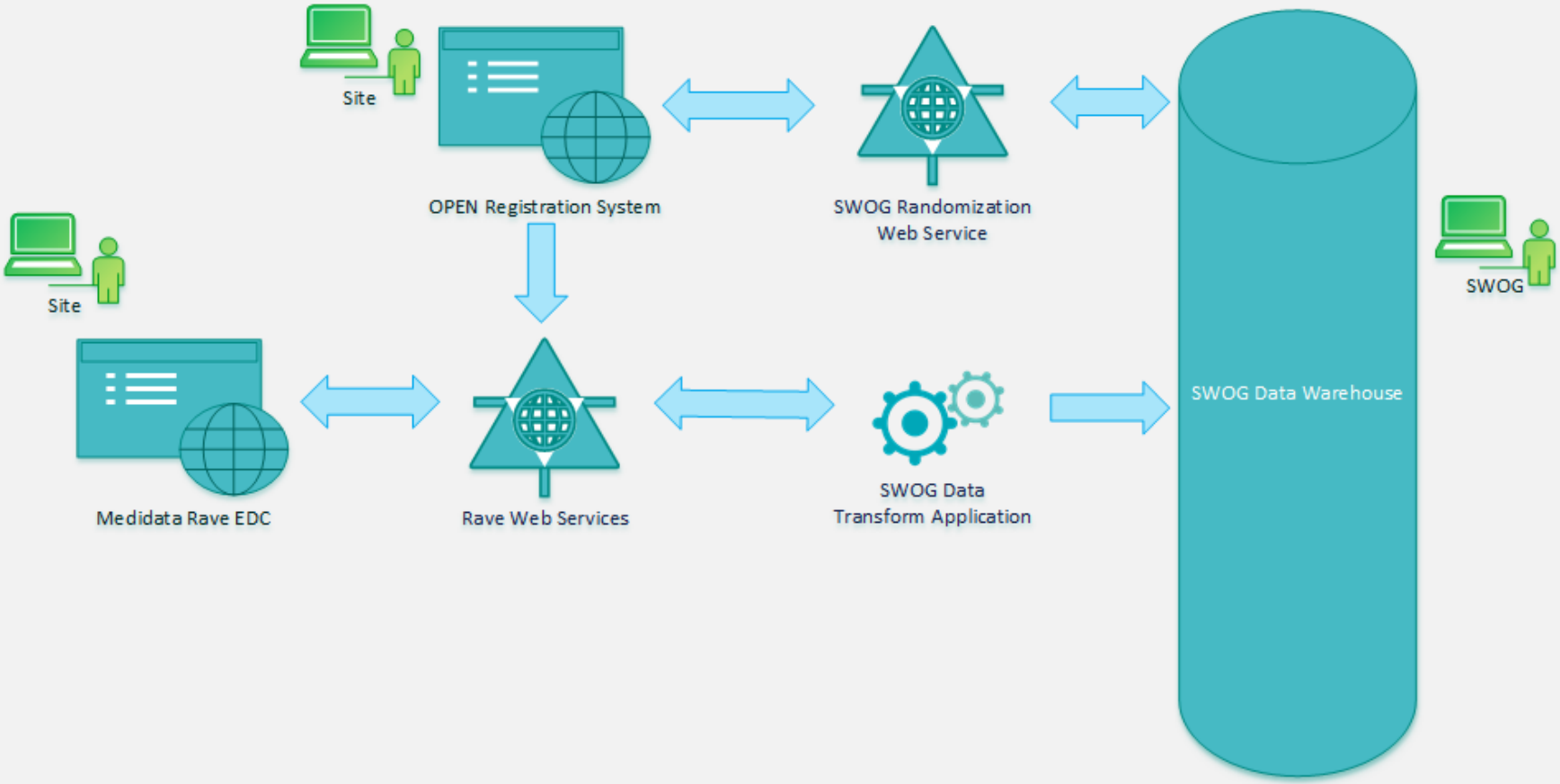


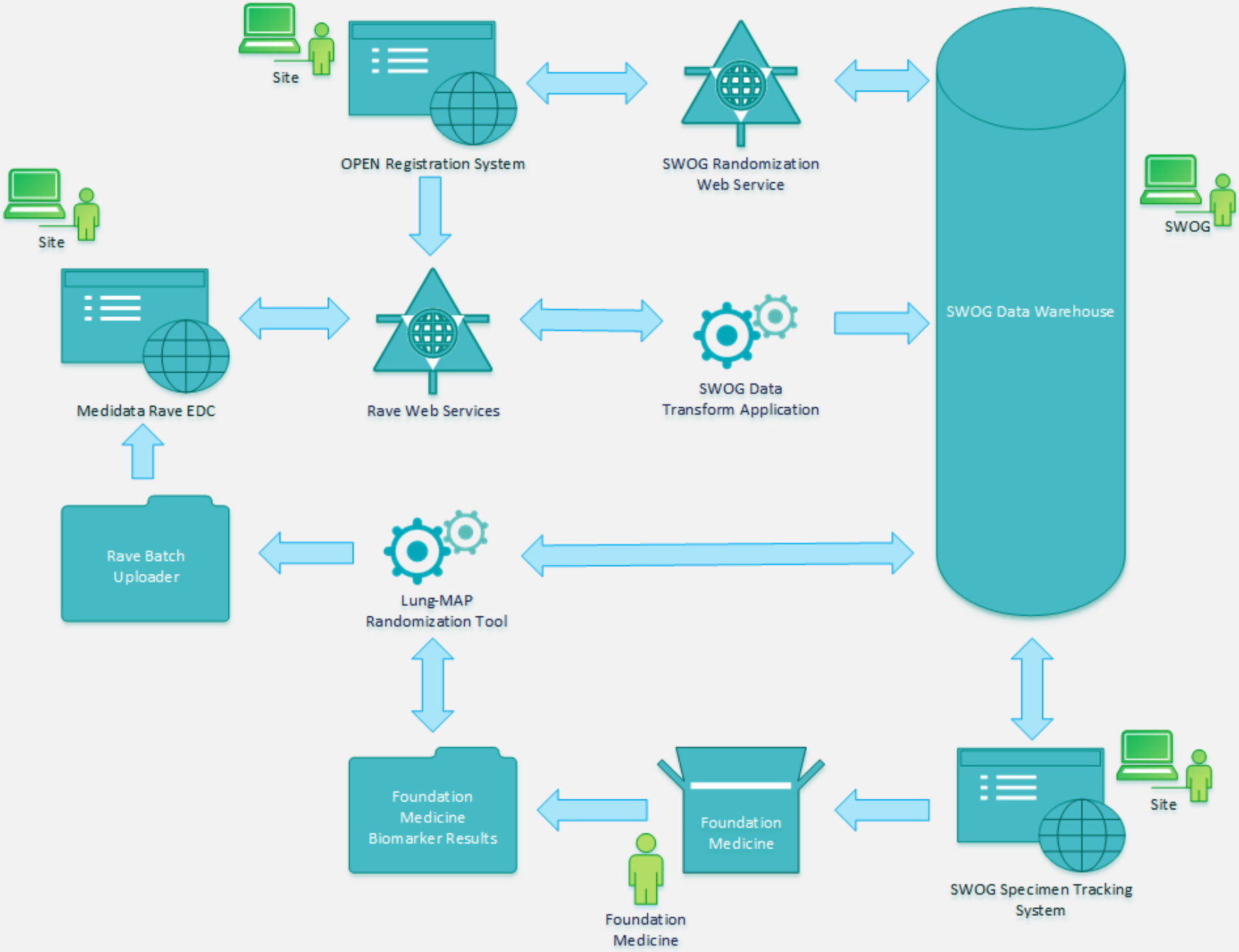
SOFTWARE REQUIREMENTS

- Unique study design brings with it unique software requirements and challenges
- The randomization scheme requires automated integration of multiple systems
  - Patient registration system (OPEN hosted by CTSU)
  - Electronic data capture (Rave hosted by Medidata)
  - Biomarker analysis (Foundation Medicine)
  - Specimen tracking application (SpecTrack hosted by SWOG)
  - Reporting and analysis tools (hosted by SWOG)
  - Final solution involved integration with 11 different components hosted by 4 different organizations.

SOFTWARE REQUIREMENTS

- Lung-MAP is a collection of trials
  - Almost 200 eCRFs
  - Coordinate same release date for all trials
- 21 CFR Part 11
- Timelines were aggressive
- Integration into current SWOG infrastructure
  - Data warehouse has over 900 studies
  - About 50 different applications interact with the data warehouse


Site
OPEN Registration System
SWOG Randomization
Web Service
SWOG
Site
Medidata Rave EDC
Rave Web Services
SWOG Data
Transform Application
SWOG Data Warehouse


Site
OPEN Registration System
SWOG Randomization Web Service
SWOG
Site
Medidata Rave EDC
Rave Web Services
SWOG Data Transform Application
SWOG Data Warehouse
Rave Batch Uploader
Lung-MAP Randomization Tool
Foundation Medicine Biomarker Results
Foundation Medicine
Foundation Medicine
Site
SWOG Specimen Tracking System

TECHNOLOGIES USED

- Web app -> Web service/XML -> Oracle
- SQL Server -> Web service/XML -> Windows application (C#) -> Oracle
- SFTP -> Oracle
- Oracle -> SFTP
- Rave Batch Uploader (SFTP) -> Rave EDC

BIGGEST CHALLENGES

- Integration – Making everything work together
  - Each of the systems wasn't that difficult to implement individually
  - 11 different systems with automation hosted by 4 different groups
  - Design failover mechanisms
- Working within aggressive timelines

LESSONS LEARNED

- Break up the work when you can
- Reuse what you can
- Set deadlines based on what you can do and not what you want to do
- Be flexible
  - Different APIs
  - Different technologies
- Build a great team