MICHIGAN STATE UNIVERSITY

Project Plan
Project Plato – GM DevBot

The Capstone Experience

Team GM

Colin Coppersmith
Simeon Goolsby
Alex Lepird
Matthew Eaton
Tao Tao

Department of Computer Science and Engineering
Michigan State University

Spring 2018

From Students…
…to Professionals

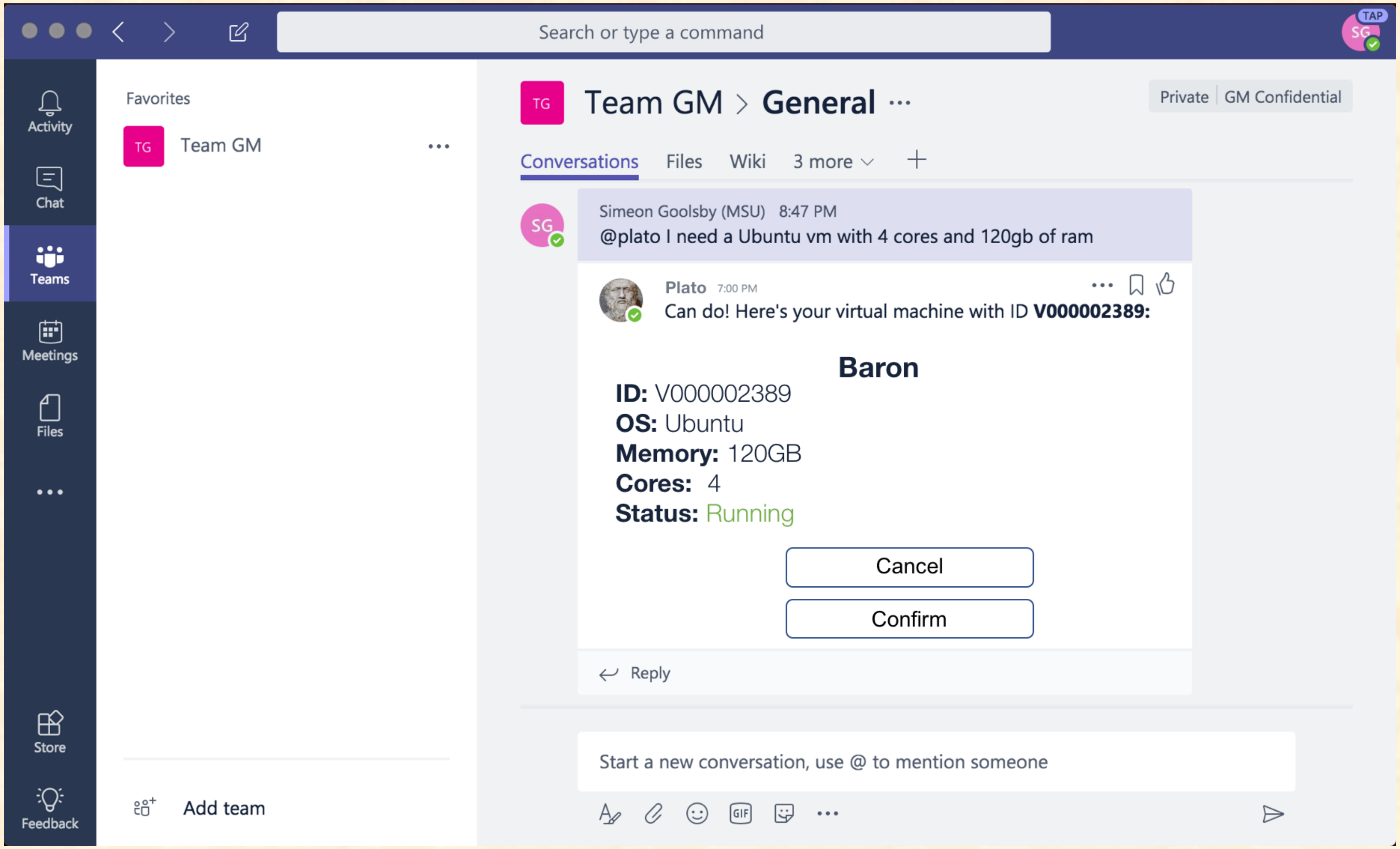
Functional Specifications

- Consolidate common developer tasks into Microsoft Teams
- Create and manipulate virtual machines
- Discover and run test cases
- Consists of three Components:
  1. AI Chatbot
  2. Microsoft Teams Tab
  3. Microsoft Teams Connector
- Provide a unified and familiar environment for Developers

Design Specifications

- Bot and Connector UI embedded in Teams chat channel
- Tab will utilize GM style guidelines from past VM management system for coloring and formatting (CSS)
- Tab consists of two sections, VM and Testing
- Tab is a high-level view of system
- Tab displays metrics and visualizations about assets
- Tab will provide UI to stop/start VM's and run tests
- Bot << extends >> Tab
- Bot integrated as a member of a team
- Bot invoked by team member saying "@plato"

Screen Mockup: VM Creation (Bot)



Simeon Goolsby (MSU) 8:47 PM
@plato I need a Ubuntu vm with 4 cores and 120gb of ram

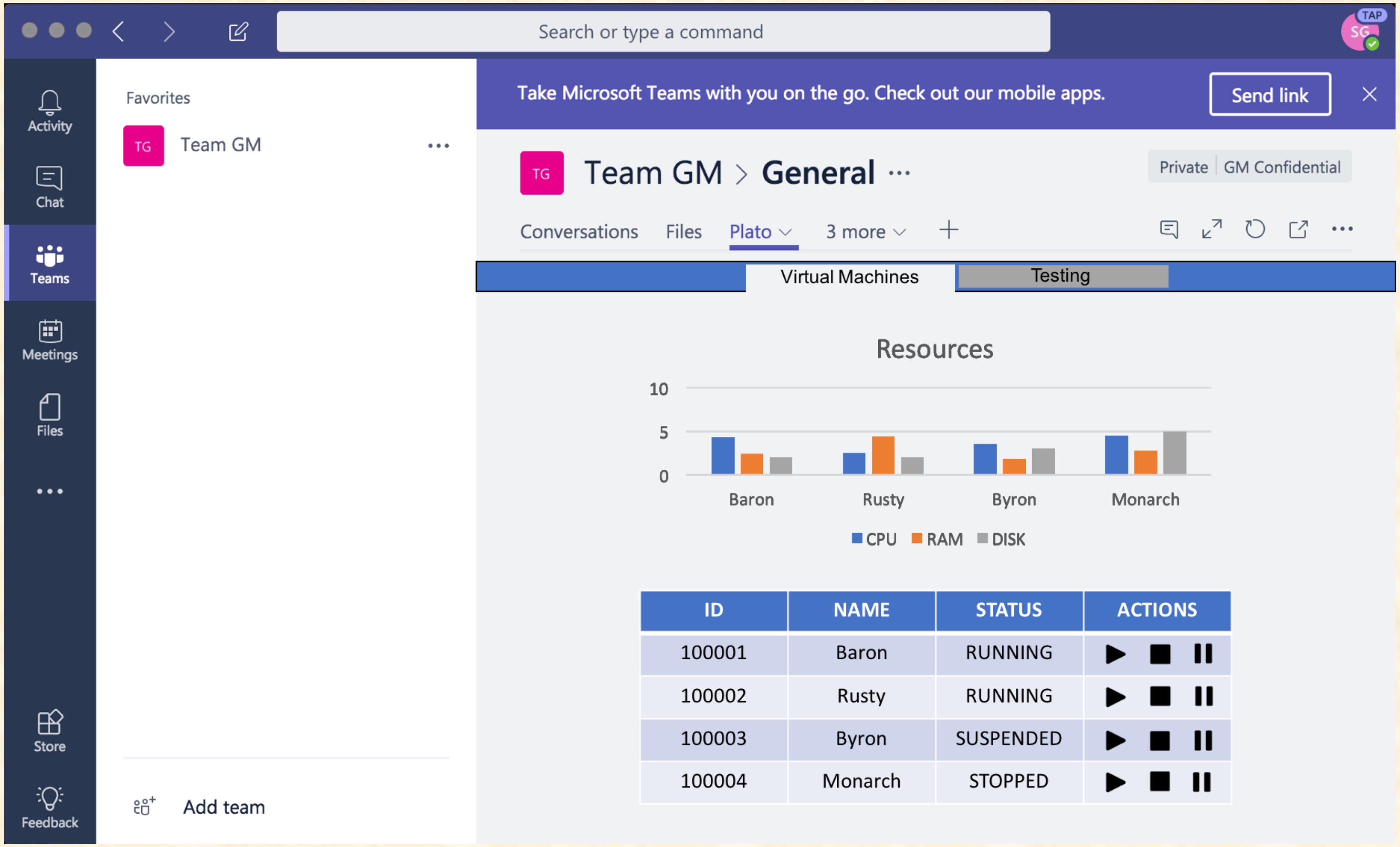
Plato 7:00 PM
Can do! Here's your virtual machine with ID **V000002389:**

Baron
ID: V000002389
OS: Ubuntu
Memory: 120GB
Cores: 4
Status: Running

Cancel

Confirm

Screen Mockup: Web Dashboard App (Tab)



Take Microsoft Teams with you on the go. Check out our mobile apps.

Team GM > General

Virtual Machines | Testing

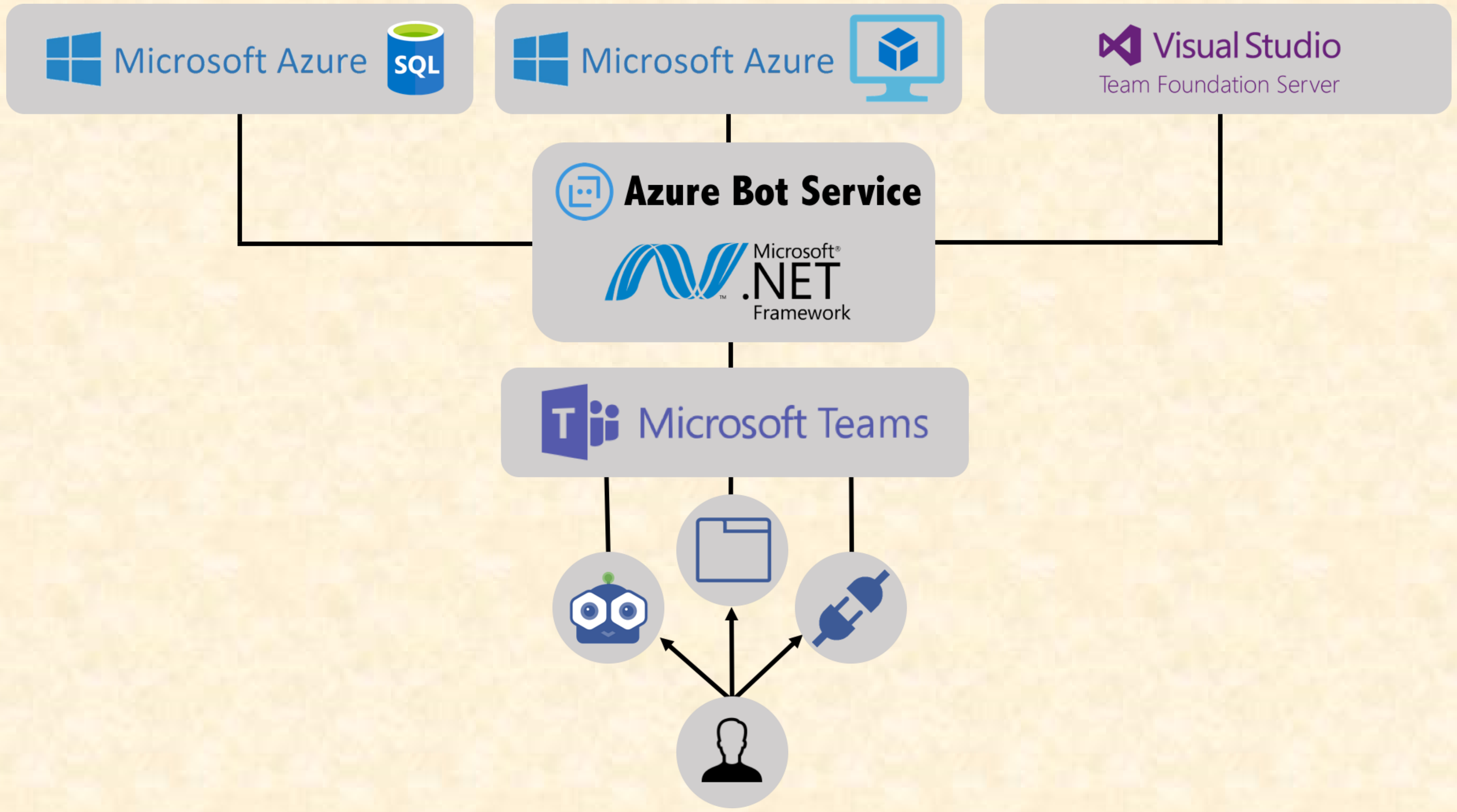
Resources

ID	NAME	STATUS	ACTIONS
100001	Baron	RUNNING	▶ ■ ❚❚
100002	Rusty	RUNNING	▶ ■ ❚❚
100003	Byron	SUSPENDED	▶ ■ ❚❚
100004	Monarch	STOPPED	▶ ■ ❚❚

Technical Specifications

- Uses Azure REST API for VM management
- Uses Test Management API for TFS
- Using Microsoft Bot Framework and LUIS for bot
- Angular.js, HTML 5, CSS, JavaScript for tab frontend
- Common C# backend controller that will communicate with Azure, SQL Server, and TFS
- Side-loading bot and web application into Teams
- Dashboard/Tab is standalone

System Architecture



System Components

- Hardware Platforms
  - Ubuntu Server on Server Rack
  - MSSQL Server Standard 2017 Database
- Software Platforms / Technologies
  - Visual Studio 2017 IDE
  - SQL Operations Studio/SQL Server Management Studio
  - Microsoft Team Foundation Server for testing, source control, and version control
  - Microsoft Bot Emulator for bot testing
  - Microsoft Teams for collaboration, testing, and deployment

Risks

- Resource Limitations
  - Description: Azure gives us $200, hosting and testing may get expensive.
  - Mitigation: Each team member can make account, can utilize other services if necessary, and could set up our own server in worst case.
- Responsive Design
  - Description: Teams has mobile app, need to adapt tab to different screen sizes and experiences.
  - Mitigation: Use responsive CSS, allow for scaling.
- User/Team Individual Experience
  - Description: Tailor experience to team and user.
  - Mitigation: Using Email as primary identifier, records in database.
- Asset Checking with Input Validation
  - Description: User could abuse bot, limited resources.
  - Mitigation: Validate against records, put upper limit on creation.

Questions?

? ? ? ?

? ?

? ? ?