Dealing with Dark Debt
Lessons learnt at Goldman Sachs

Vanessa Yiu
Site Reliability Engineering, Goldman Sachs
SREcon EMEA 2018
Agenda

- SRE@GS
- What is “Dark Debt”?
- How to combat Dark Debt
- Q&A
SRE@GS: Who are we?

Leading global financial services firm
- Investment banking
- Securities services
- Investment management services

Client base includes
- Corporations
- Financial institutions
- Governments
- Individuals

Founded in 1869 as a commercial paper provider

~38k employees in over 30 countries
SRE@GS: Who are we?

One in three of our employees are engineers
SRE@GS: Scale and impact

Why are we at SREcon?

1.5bn+ lines of code

~7k software products / applications

200k compute nodes

50TB data processed daily in our DataLake
SRE@GS: Scale and impact

Why are we at SREcon?

Managing system risks and complexity at enterprise scale is the core of what we do as SREs at Goldman Sachs
“Dark debt is found in complex systems and the anomalies it generates are complex system failures. Dark debt is not recognizable at the time of creation. ... It arises from the unforeseen interactions of hardware or software with other parts of the framework. ...”

STELLA Report from SNAFUcatchers Workshop on Coping With Complexity (2017)
How to combat Dark Debt

1. PREVENTION
2. INSIGHT
3. DETECTION
4. DIAGNOSE
5. CULTURE
How to combat Dark Debt

1. Build sustainable development ecosystems.
SecDb Ecosystem

• Central risk management platform for Goldman Sachs
• Object oriented database with dependency graph
• Has own language and integrated development environment
• Invented 25 years ago but only 2 weeks old!
SecDb Ecosystem

- Designed with transparency and collaboration from the outset
- Everyone can read code and see dependencies
- Everyone can write and execute code locally too!
SecDb Ecosystem

• Heavy investment in linting and beaconing

Examples:

• Instrumentation of complete runtime environment

• Automated bots

• “Fix Me!” comments

• Gamification to improve codebase
How to combat Dark Debt

**INSIGHT**

Decouple monitoring tools from application.
Java Runtime Monitor (JRM)

- Agent embedded into JRE
- JMX metrics
- Beaconing / instrumentation
- Developers can expose app specific data
- All data ingested into centralised monitoring / alerting / data analytics framework
- Configuration **completely decoupled** from sourcecode
How to combat Dark Debt

Controlled chaos is good for your systems.
Chaos Engineering

• Controlled experiments to test for anomalies in system behaviour
• Flags issues with potential to cause outages
• Helps uncover systemic weaknesses, including dark debt, \textit{at point in time}
• But systems are not static!
• Recalibrate and run continuously across environment
How to combat Dark Debt

1. PREVENTION
2. INSIGHT
3. DETECTION
4. DIAGNOSE
5. CULTURE

**Visualise** issues in your distributed systems.
Distributed Tracing
Distributed Tracing

Span

Trace
Distributed Tracing
Distributed Tracing
How to combat Dark Debt

Help *foster right culture and practices.*
Dos and Don’ts
The Don’ts
The Don’ts

Don’t play Whack-A-Mole!
The Don’ts

Say no to “dark matter” developers
The Don’ts

Don’t ignore your technical debt
The Dos

[Images of various icons, possibly representing different concepts or actions related to dos and don'ts]
The Dos

Increase runtime transparency
Practice blameless post-mortems
Pair programming and peer code reviews

The Dos
The Dos

SRE hackathons / dedicated sprints for refactoring
How to combat Dark Debt

1. **PREVENTION**
   - Build *sustainable* development ecosystems.

2. **INSIGHT**
   - *Decouple* monitoring tools from application.

3. **DETECTION**
   - *Controlled chaos* is good for your systems.

4. **DIAGNOSE**
   - *Visualise* issues in your distributed systems.

5. **CULTURE**
   - Help *foster right culture and practices*. 
Make things possible.

JOIN OUR SRE TEAM

Contact us at sre-hiring@gs.com