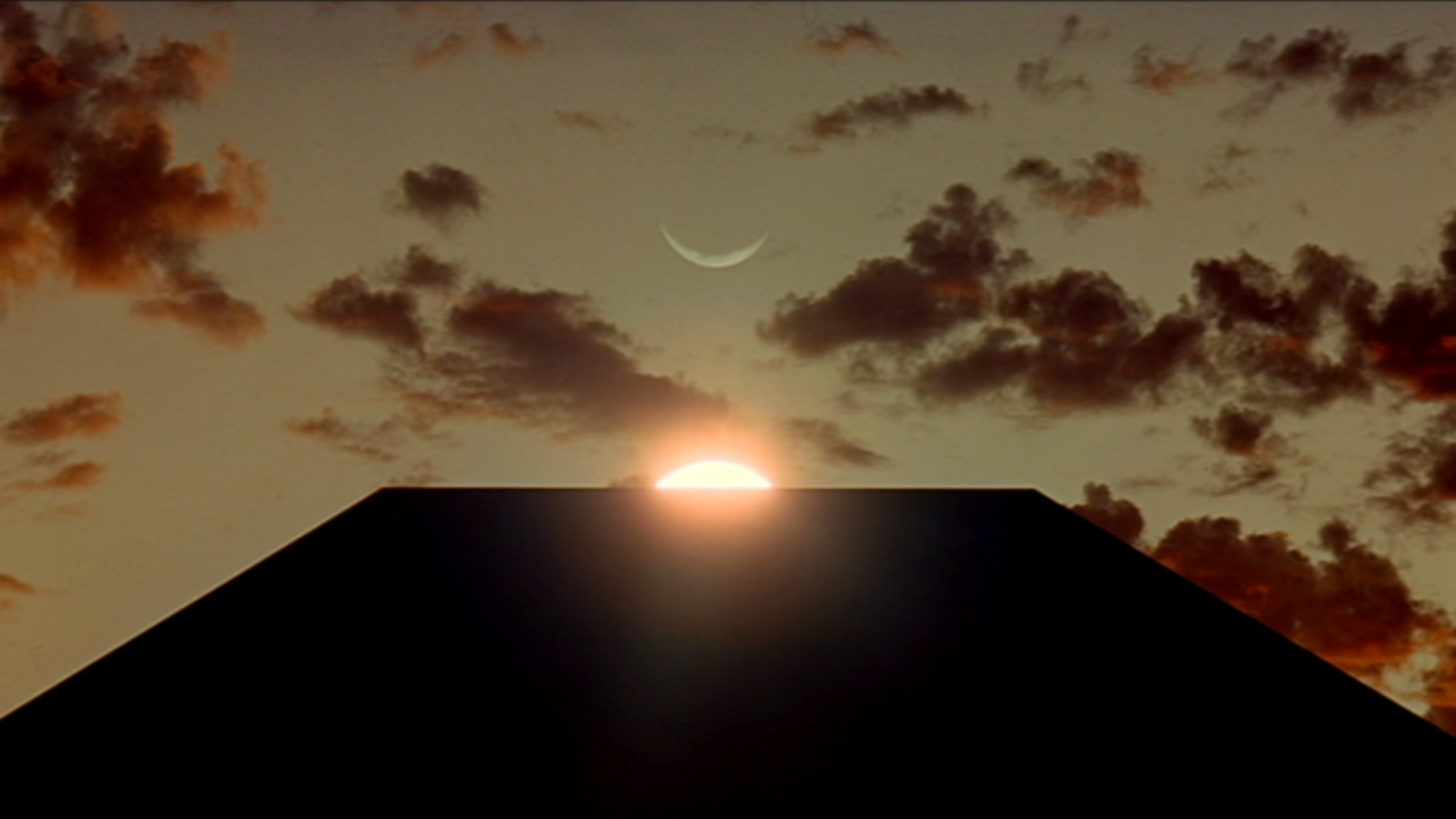


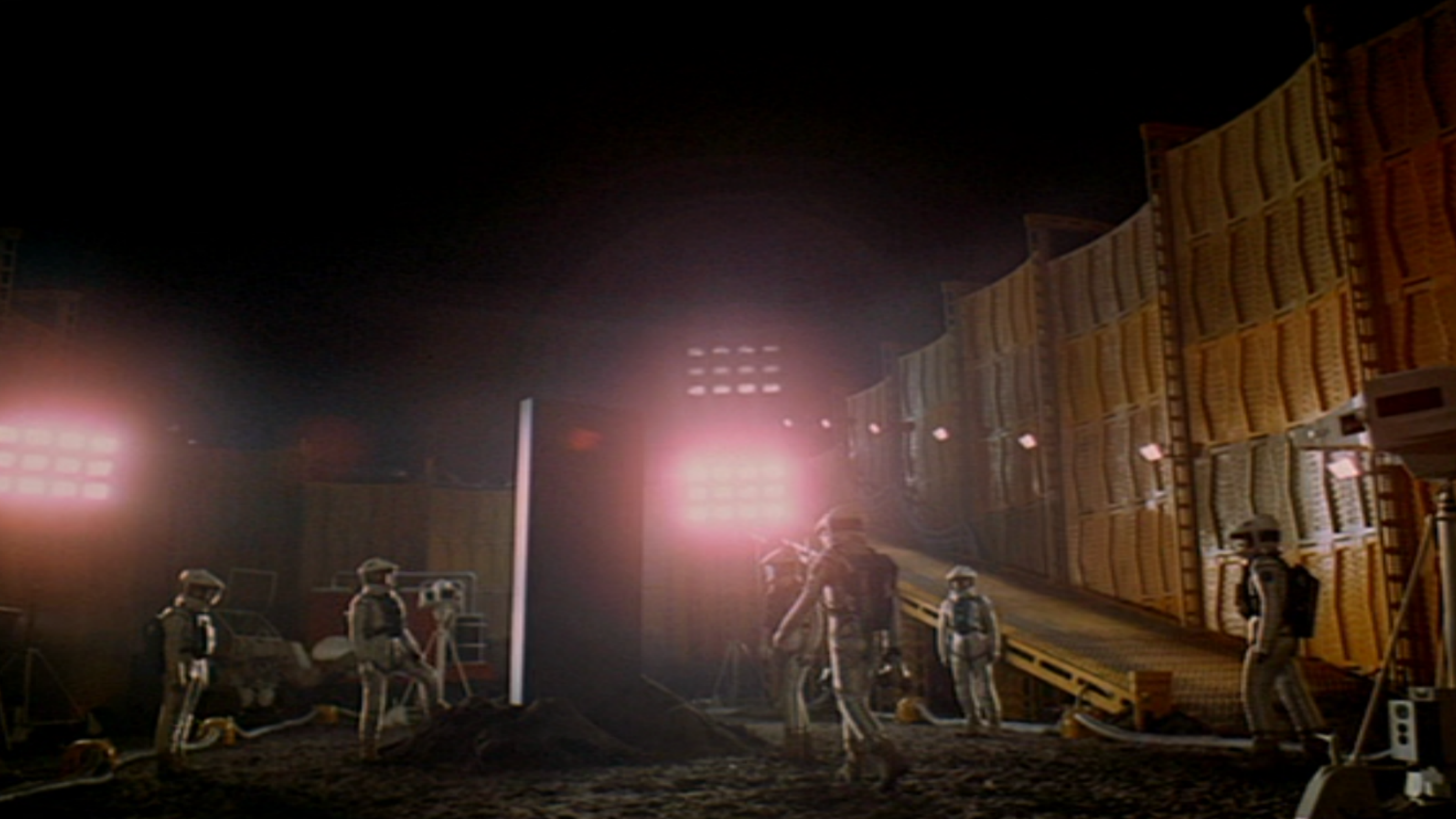


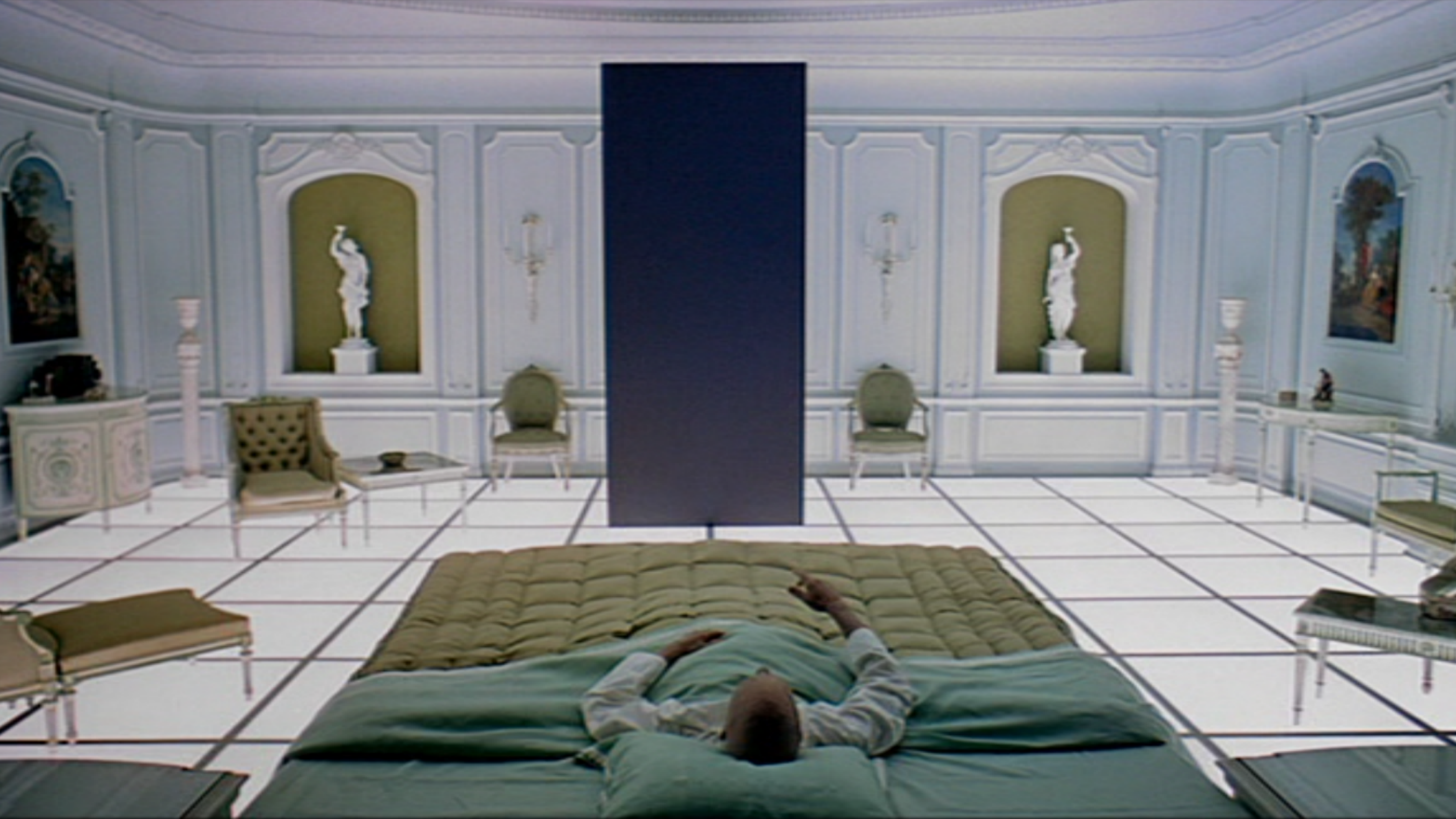
2001: A SPACE ODYSSEY

© MCMXLXVIII by Metro-Goldwyn-Mayer Inc. All rights in this Motion Picture Reserved Under International Conventions









**Monoliths,
Balls of Mud
& Silver Bullets**

Monoliths
are bad

Monolithic
Balls of Mud
are worse



Microservices
will save us

Well...

I am old

Old people have
long memories

“Those who refuse to learn from commit history are doomed to reimplement it”

ANNIVERSARY EDITION WITH FOUR NEW CHAPTERS



ESSAYS ON SOFTWARE ENGINEERING

THE MYTHICAL MAN-MONTH

FREDERICK P. BROOKS, JR.

“Adding manpower
to a late
software project
makes it later.”

“Observe that for the programmer, as for the chef, the urgency of the patron may govern the scheduled completion of the task, but it cannot govern the actual completion.”

“There is
no silver bullet”

“No Silver
Bullet”

HALF-MAN... HALF-WOLF

The
CURSE
OF THE

WEREWOLF

TECHNICOLOR®

CERT X ADULTS ONLY

CLIFFORD EVANS · OLIVER REED · YVONNE ROMAIN · CATHERINE FELLER



Screenplay by

JOHN ELDER

Directed by

TERENCE FISHER

Produced by

ANTHONY HINDS

Executive Producer

MICHAEL CARRERAS

A GAMMA FILM PRODUCTION by UNIVERSAL INTERNATIONAL



Microservices
are not a
Silver Bullet

Disclaimer

Please note that nothing I say over the rest of this presentation should be taken as me saying or implying that microservices are not an improvement over most software architectures that preceded them. I am merely pointing out that they are not the cure-all that some of their fans seem to think they are and that some care needs to be taken in order to get the most benefit from them.

I'm also, of course, going for a few cheap laughs here. All advice is worth exactly what you paid for it. Satisfaction guaranteed or your money back.

Advantages of microservices

Simplicity

~~Simplicity~~

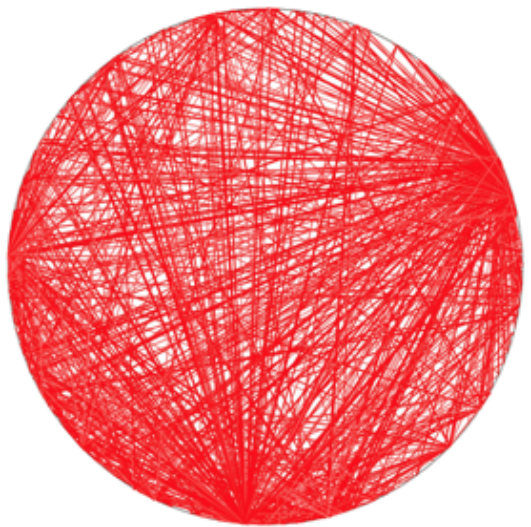
Less complexity

Ownership

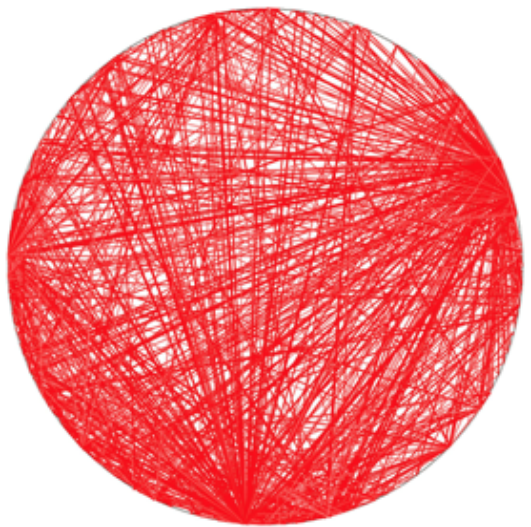
Autonomy

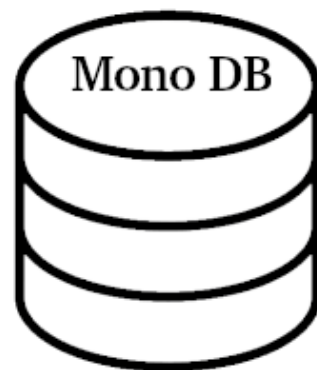
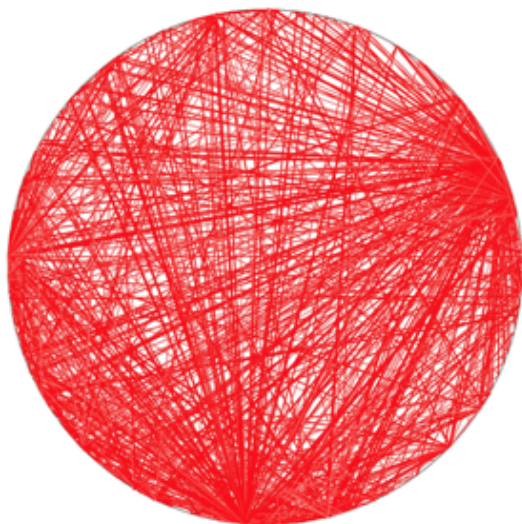
Faster releases

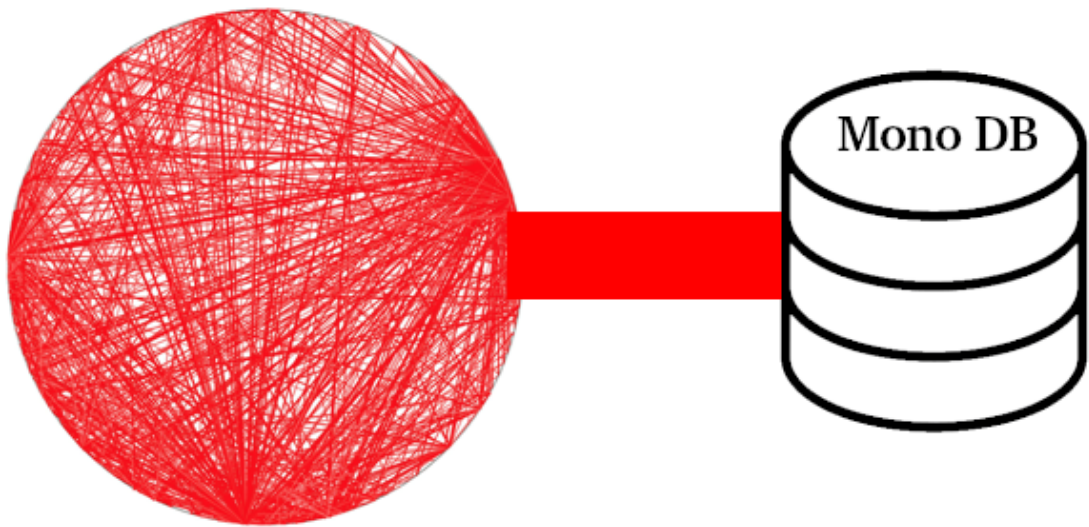
Inside your
monolithic
ball of mud





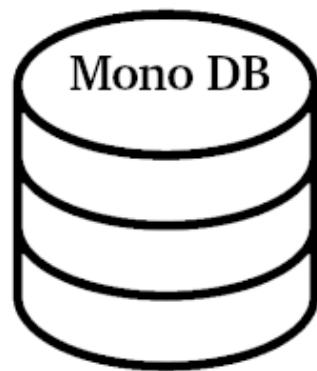


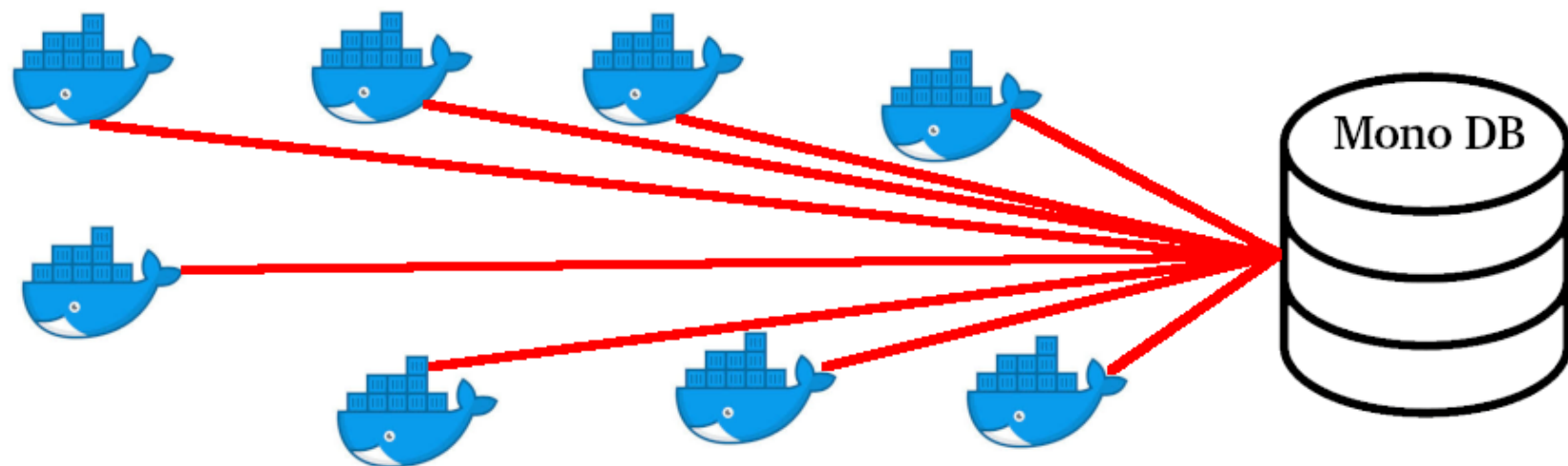




Microservices
version







Microservices on
a monolithic
database

Who owns the
database
schema?

How do you
change the
database
schema?

How do you ensure you that your changes to the database schema don't affect other users of the monolithic database?

Microservices
need
microdatabases

Simplicity?

Ownership?

How many stacks
do you need?



Dev



Dev

QA



Dev

QA

Integration



Dev

QA

Integration

Production

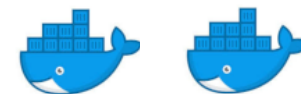
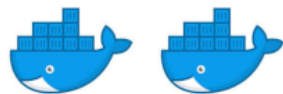
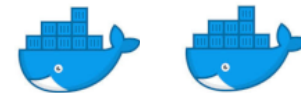
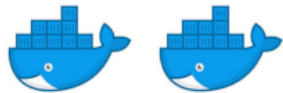
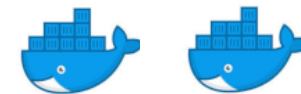
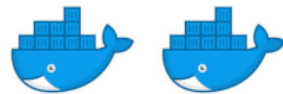
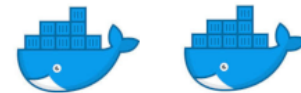
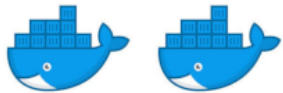


Dev

QA

Integration

Production

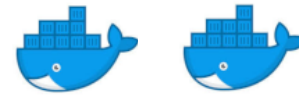
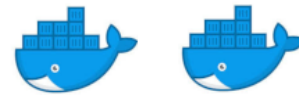


Dev

QA

Integration

Production

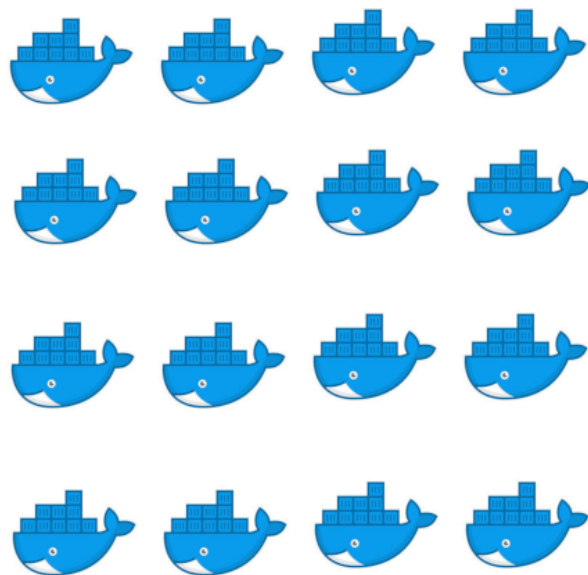


Dev

QA

Integration

Production



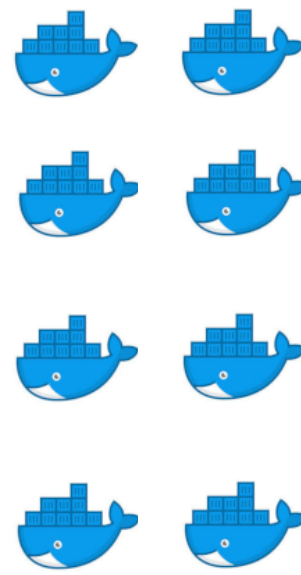
Dev



QA



Integration



Production

AWS === £\$€

Some tips

Microservices are
REST over HTTPS

<SOAP/>

Dodger

JSON is standard

HTTP is
more than
GET and POST

Use HTTP Verbs for CRUD

CRUD	REST over HTTP
Create	POST
Read	GET
Update	PUT
Delete	DELETE

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

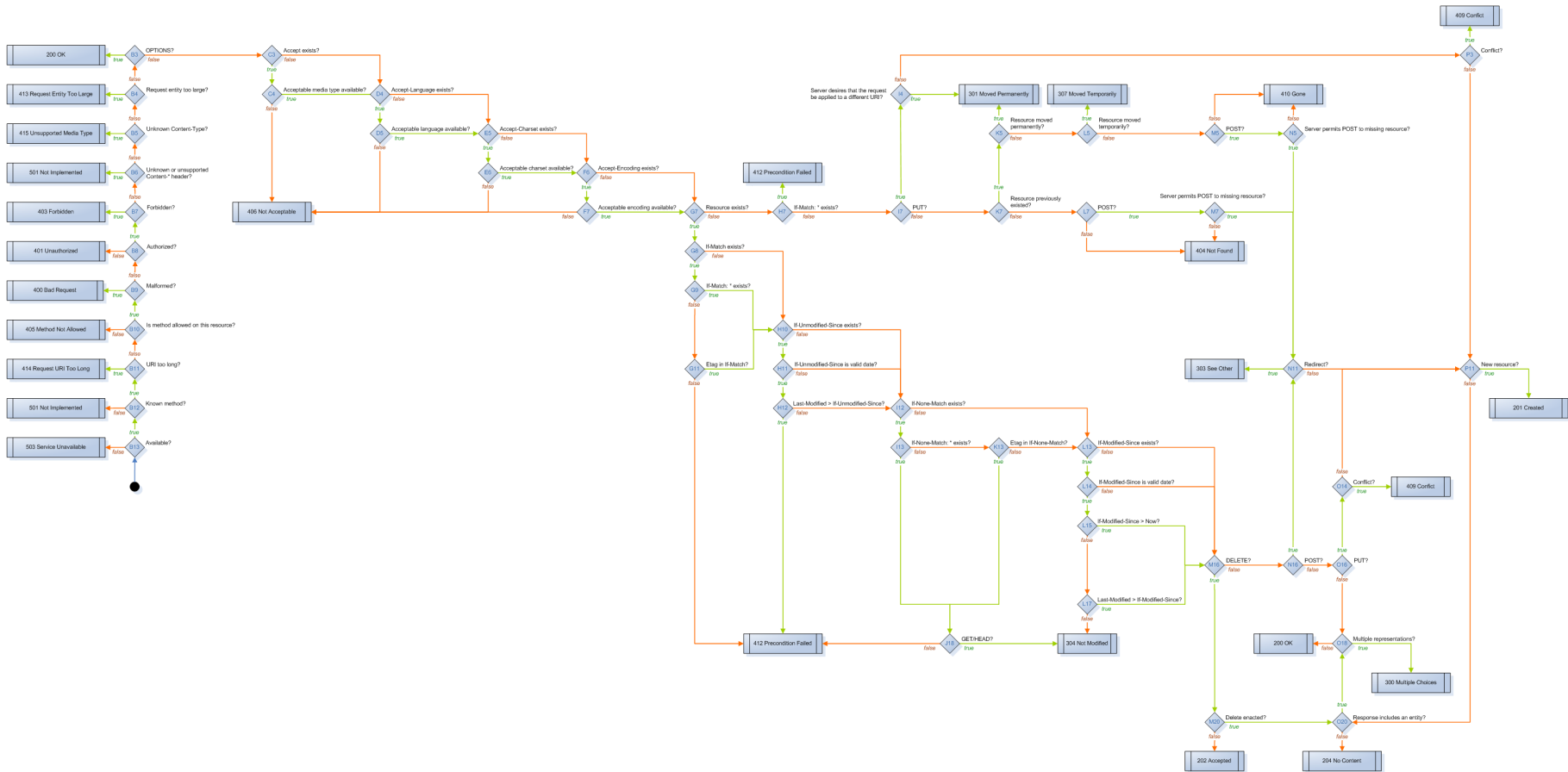
19

20

21

22

23



Assume your
private API
will be public

Issue API keys

Version your API

Use HAL

(Hypertext Application Language)

Generate docs
from specs
(Swagger)

Generate docs
from specs
(~~Swagger~~)

Generate docs
from specs
(OpenAPI)

Emulate
Johnny Cash

Emulate Johnny Cash

Destroy the monolith
“One Piece at a Time”





Dave Cross
<https://perlhacks.com>
@perlhacks