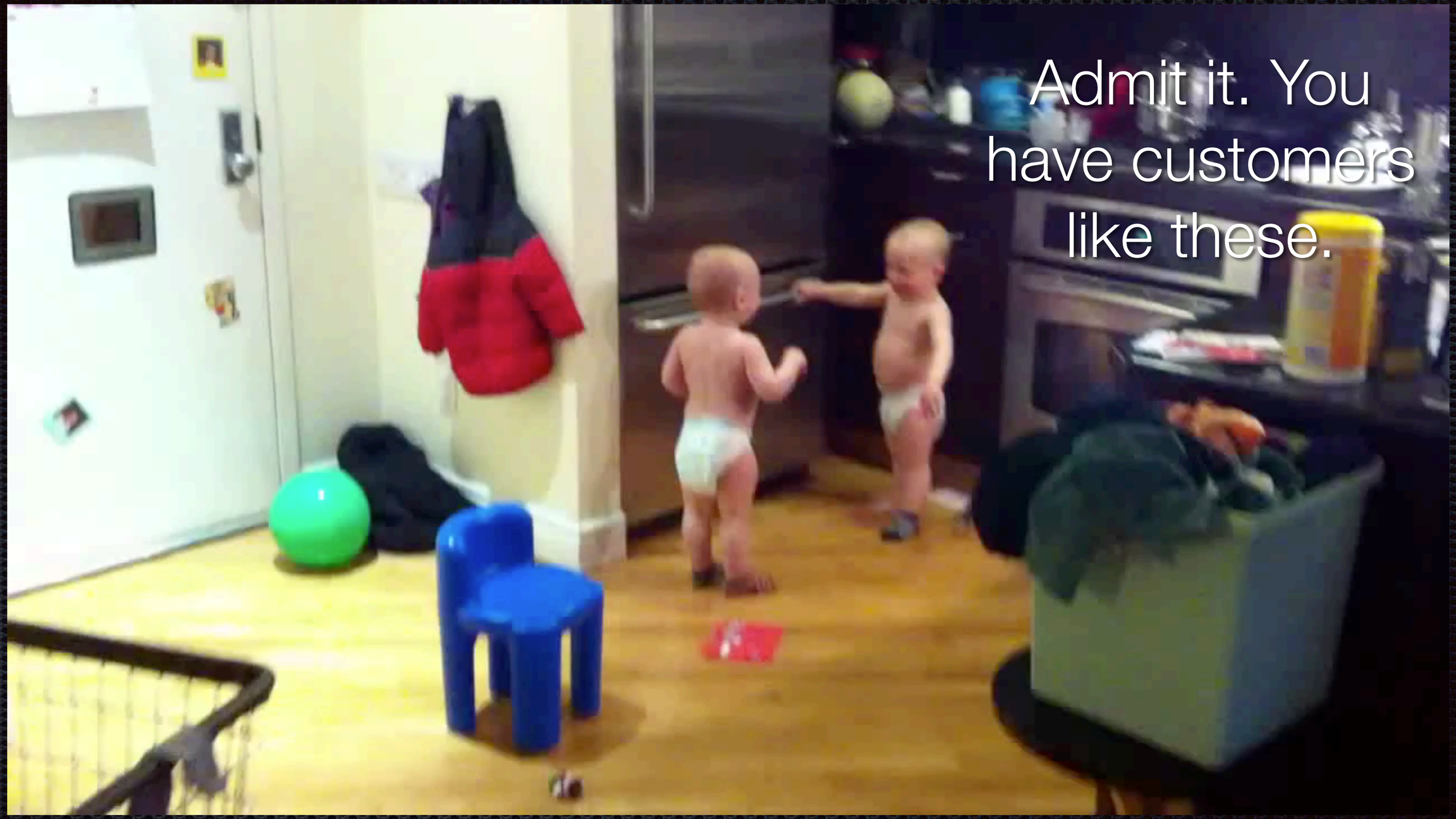
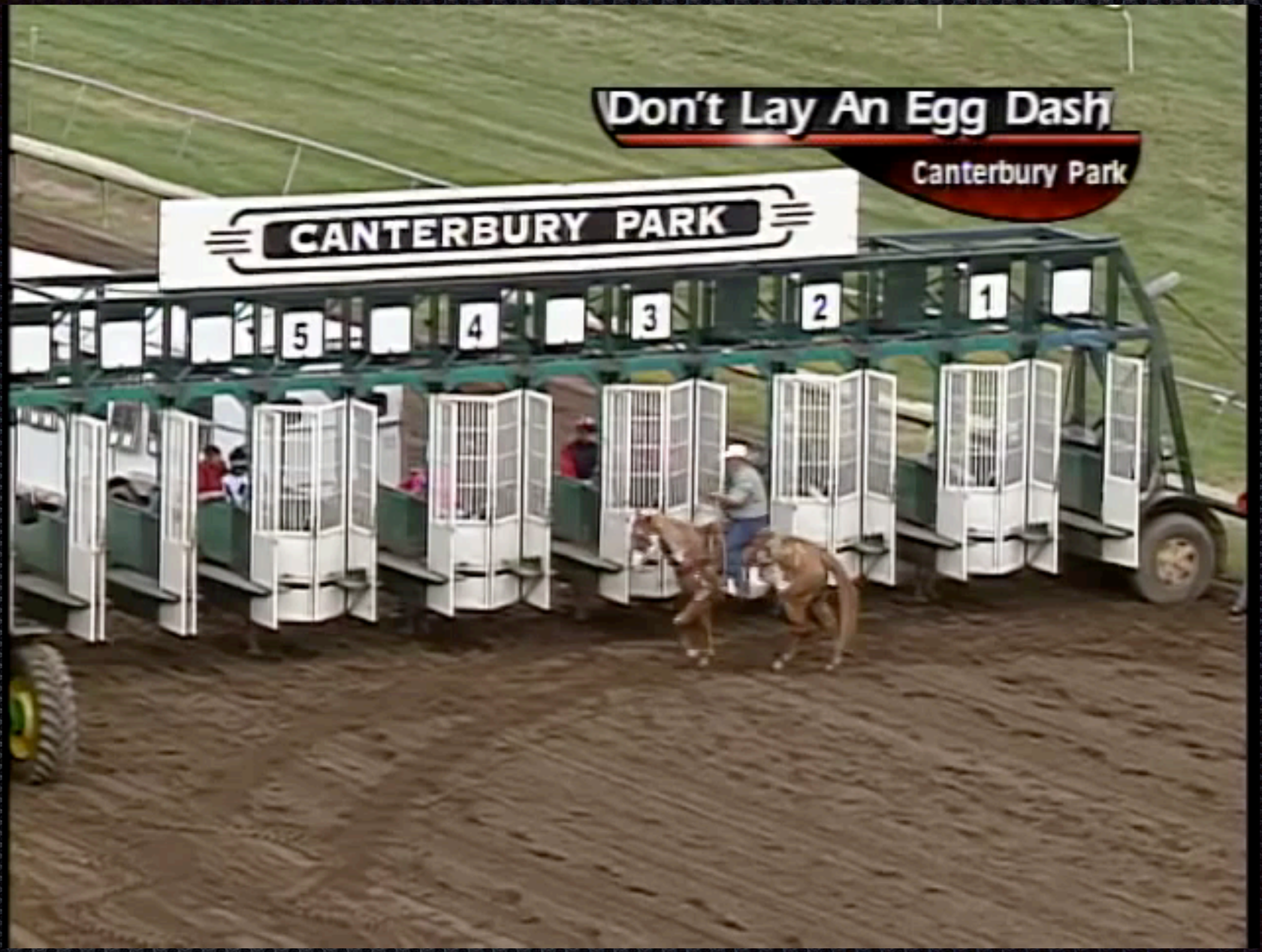

How to accelerate anything*
with ElectricAccelerator

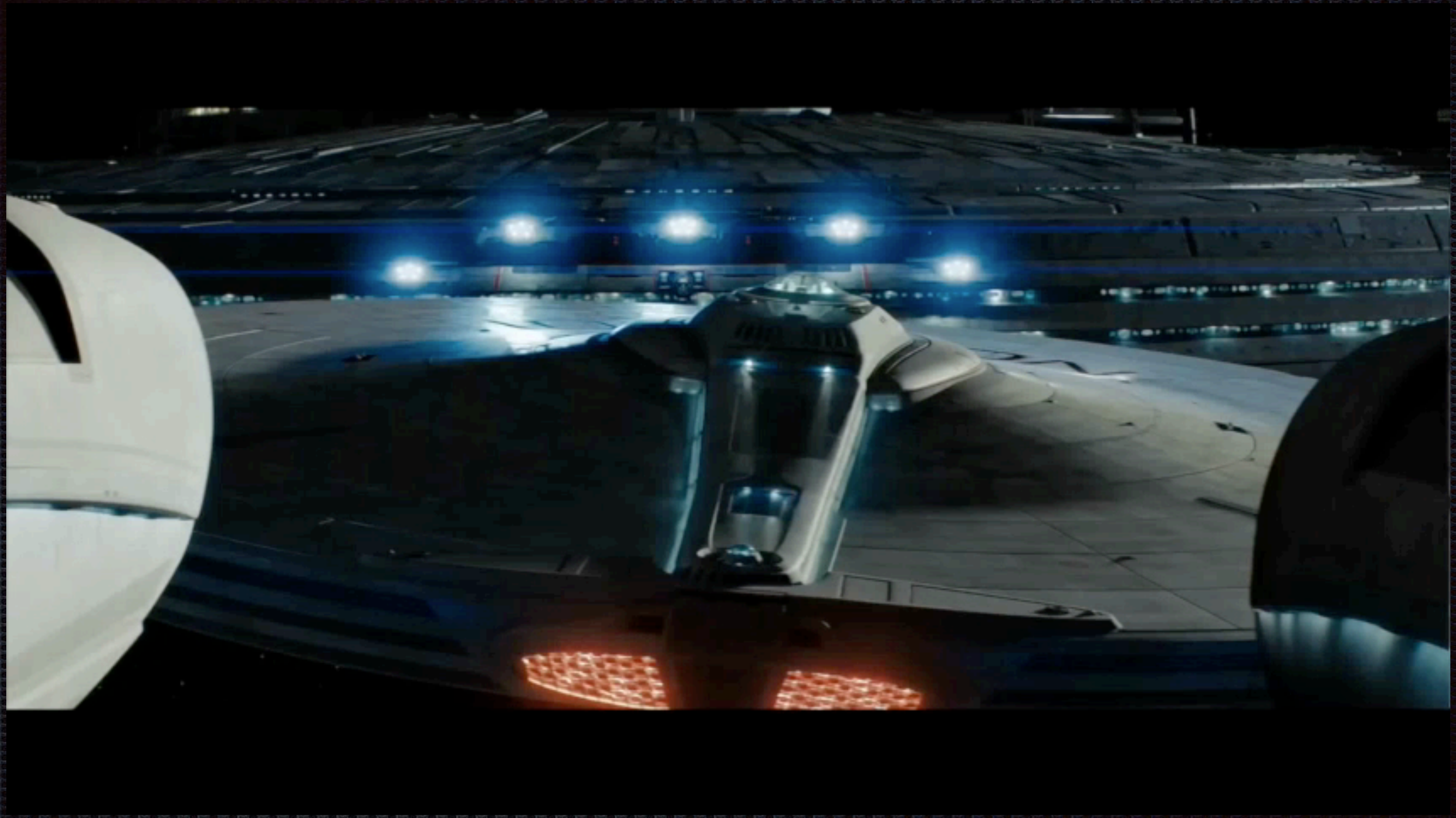
The story.


Admit it. You
have customers
like these.

The sprint.

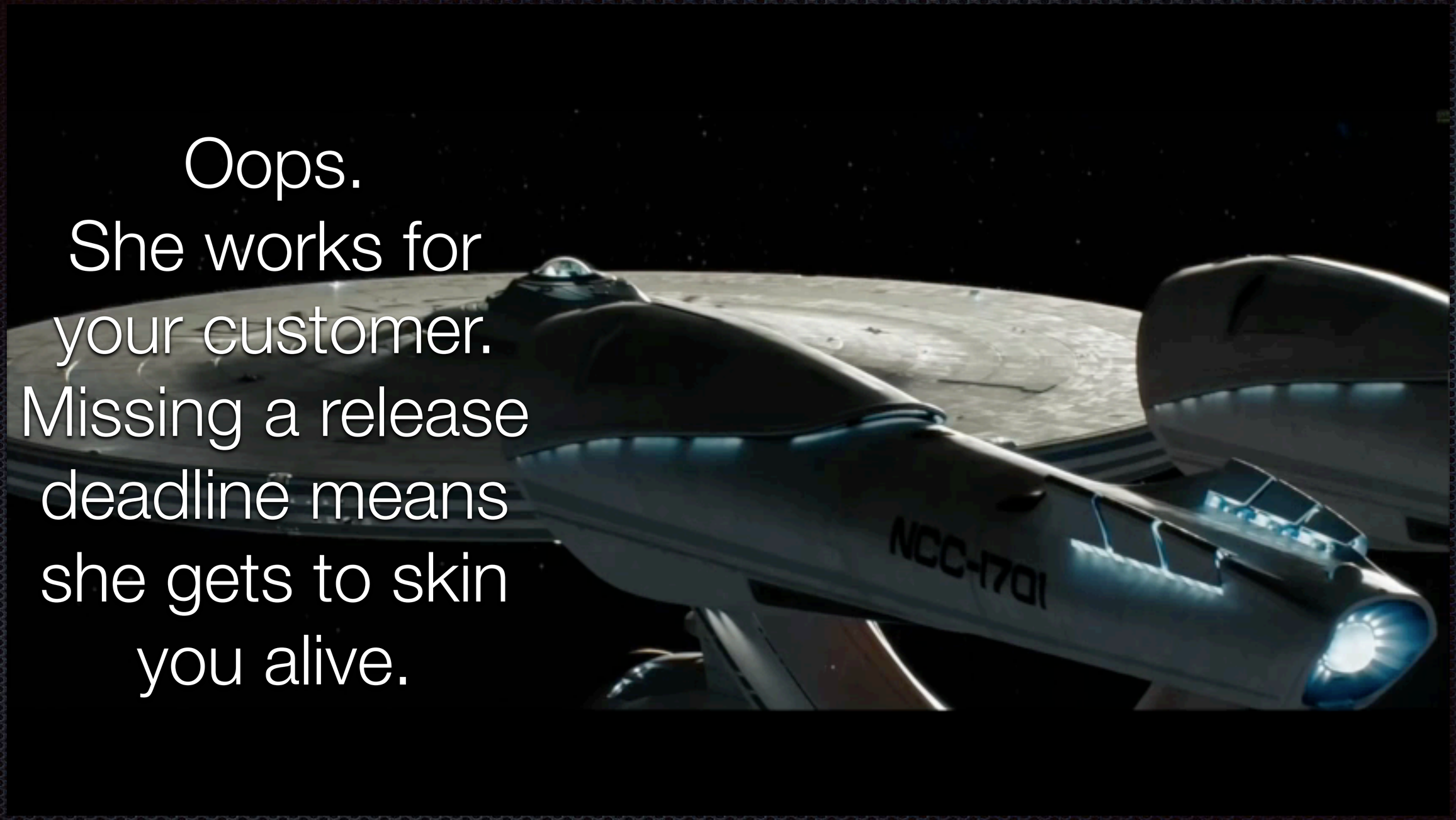


Continuous Integration....



...meets testing.

Oops.
She works for your customer.
Missing a release deadline means she gets to skin you alive.



Been there?
Done that?

#1 cause of IT failures?
Insufficient testing.
#1 cause of that?
It takes too long.
#1 consequence of that?
The humans cut corners.

The problem?

Any collection of tasks can only be as fast as the slowest link.

The solution?

1. Bigger box with more cores
2. Parallelize with many (smaller?) boxes and cores

What are the considerations?

The Granularity of the Work

How much work is done in each process?

The Problem to be Solved

Stress-testing a web site, or running a massive database analysis?

What is the goal?

Increase throughput?
Reduce turnaround time?

Case Study #1

Static pattern and flow analysis of individual projects in a Visual Studio C# solution

Iterative tasks are excellent parallelization candidates using makefiles

Makefile content:

```
PROJECTS=utils core library external graphics gui xml
TARGETS=$(addprefix go_,$(PROJECTS))
all: $(TARGETS)
go_%:
    dottestcli -nobuild -report .\project\$* -project .\project\$*\$*.csproj
```

```
emake  --emake-cm=<your_cluster_manager> --win32
```

Case Study #2

Static pattern and flow analysis of an entire C++ product

Requires cooperation from the tool to split the work into processes and parallelize with Electrify

Electrify works by intercepting new instances of processes and sending them to agents

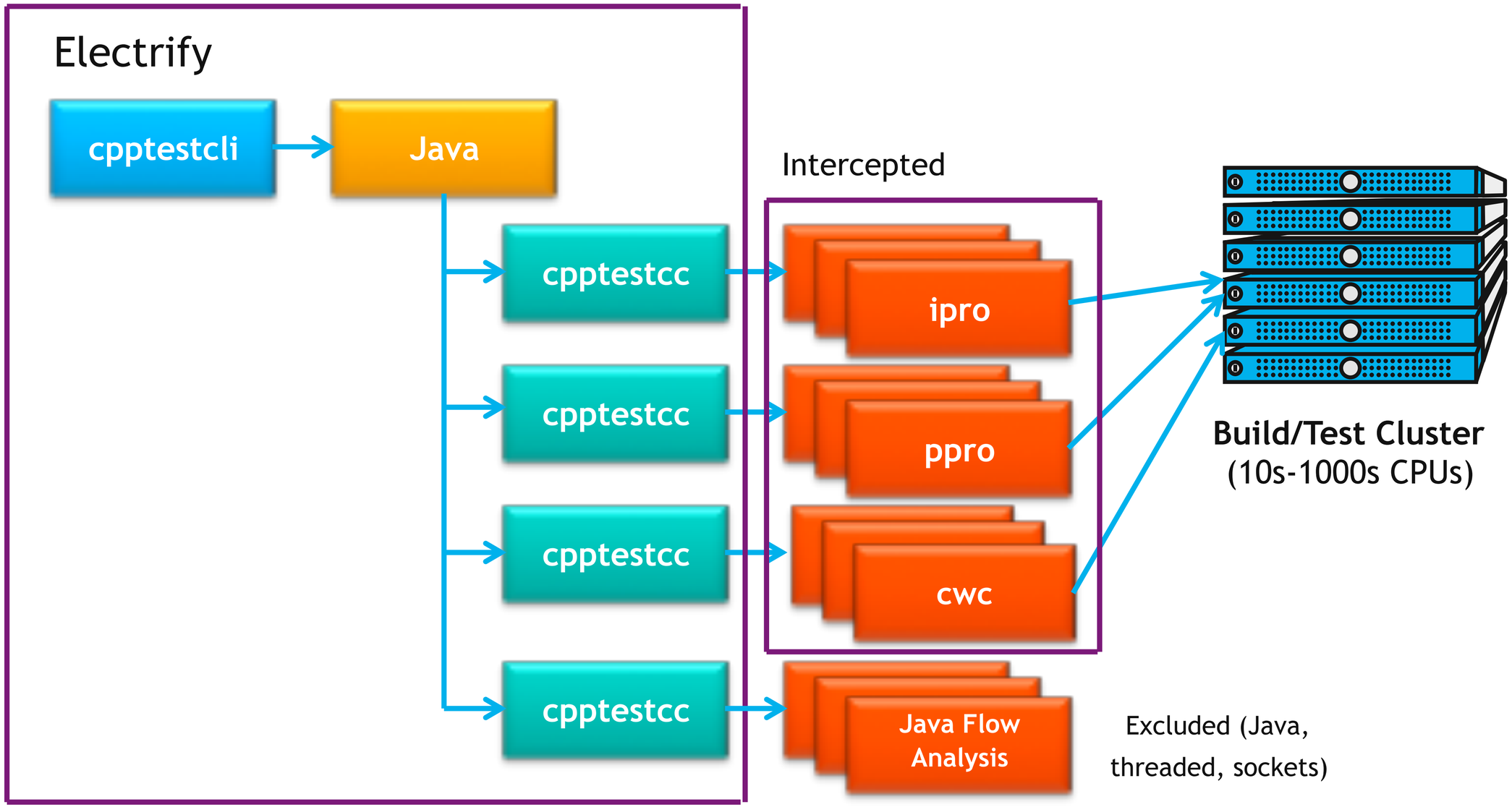
The catch? No emake.

Command line:

```
electrify  --emake-cm=<your_CM_IP_or_FQDN>
           --electrify-remote=ipro:ppro:cwc
           -- cpptestcli
              -settings cpptestcli.settings.txt -input cpptestscan.bdf
              -compiler vc_10_0
```

```
cpptest.analyzer.flow.multiprocess.analysis=true
cpptest.analyzer.flow.multiprocess.analysis.using.socket=true
parallel.mode=Manual
parallel.max_threads=6
```

How Electrify accelerates a Parasoft analysis run



Electric Cloud

Results?

8x to 12x acceleration.
8-12 hour analyses reduced to 1 hour and incorporated into CI schedule.

For you BPO's…
(Business Prevention Officers a.k.a. Bean Counters)
70 developers. $200k loaded cost.
50% productivity loss during builds/test runs.
Serialized: 1 hour full, 10 minutes CI @ 5/day.

$3,791,667

Parallelized: 10 minutes full, 1 minute CI @ 20/day.

$437,500

ROI: $3,354,167
PER YEAR. KA-CHING!

What else can be parallelized?

Stress testing.
Remember healthcare.gov?

Processing discrete datasets.
Financials.
GIS.
Engineering.
Reporting.
Dataset exports.

Architecting applications to ~~run~~ fly with ElectricAccelerator

Electric Cloud

Processes, not threads.
Discrete file I/O.
No pipes or daemons.
No inter-process dependencies.
CLI.
Configurable parallelization.

Questions?

Thank you!

jjimenez@electric-cloud.com