

Form follows function  
**Getting the right mix  
in user response monitoring**

Alistair A. Croll  
Chief Strategy Officer  
Coradiant, inc.

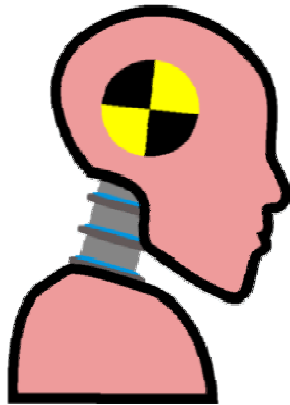
**NETWORLD  
+ INTEROP**



# Form follows function

## **Getting the right mix in user response monitoring**

Featuring...

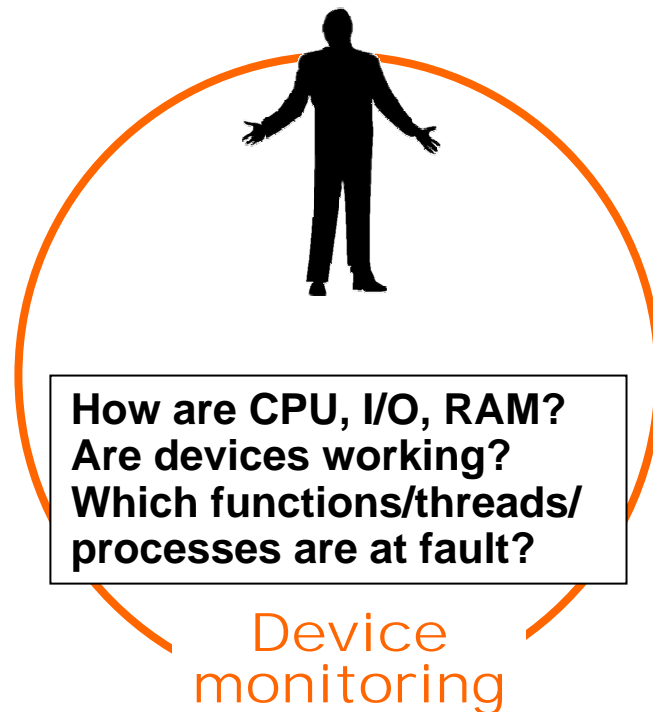


The well-behaved  
synthetic test



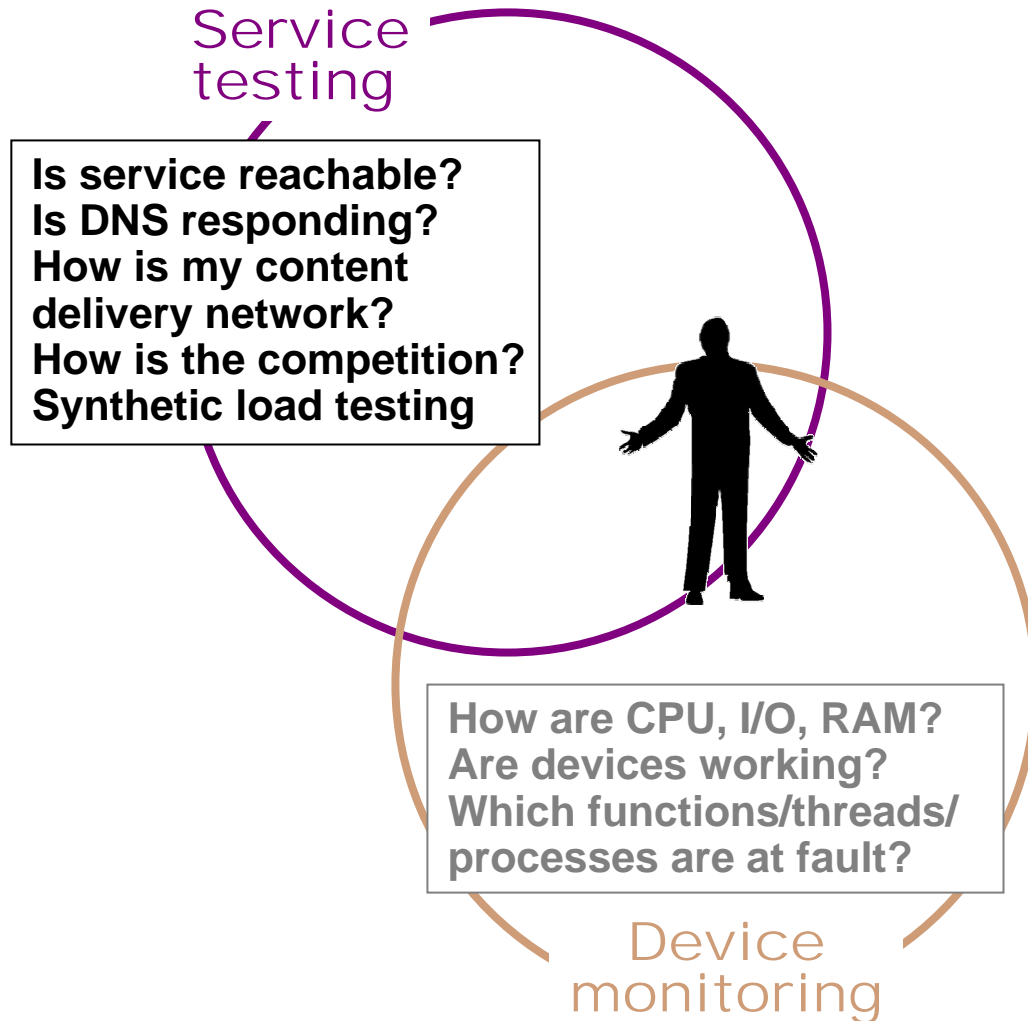
The messy, ornery  
real user traffic

# In the beginning, there was the device



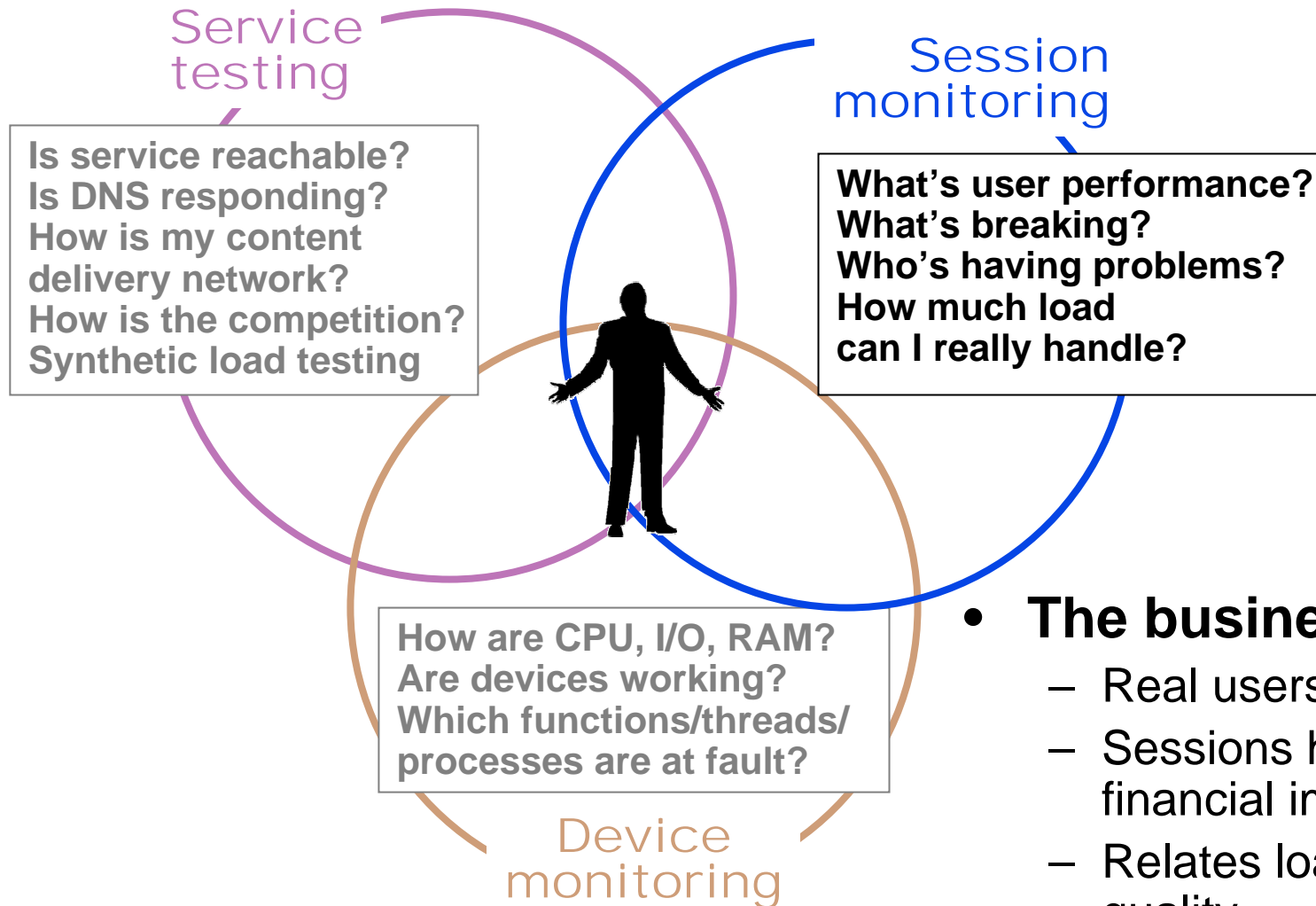
- Great for **single-vendor** environments
- If you control **end-to-end**, you can discover user experience from the infrastructure itself

# Then the world got messy: So we started watching the application



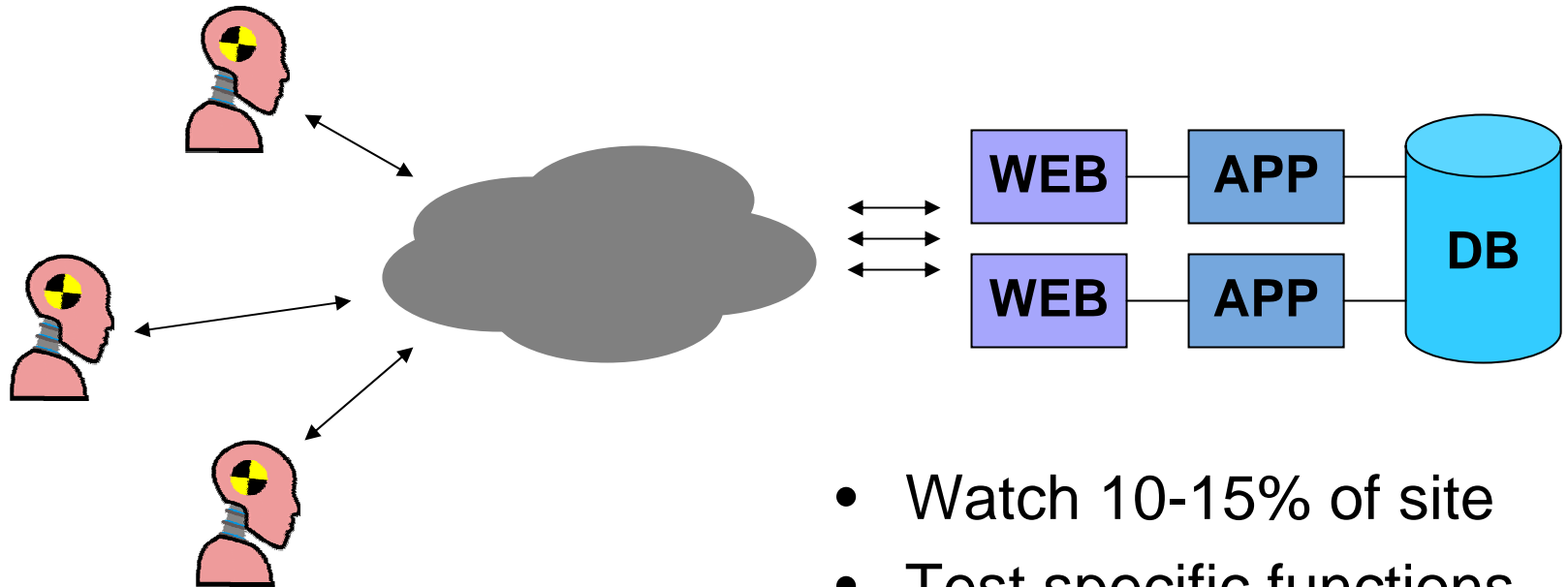
- **Mixed software** and many platforms
- Need to **test the service** to verify it's functioning

# Today: Session-centric monitoring



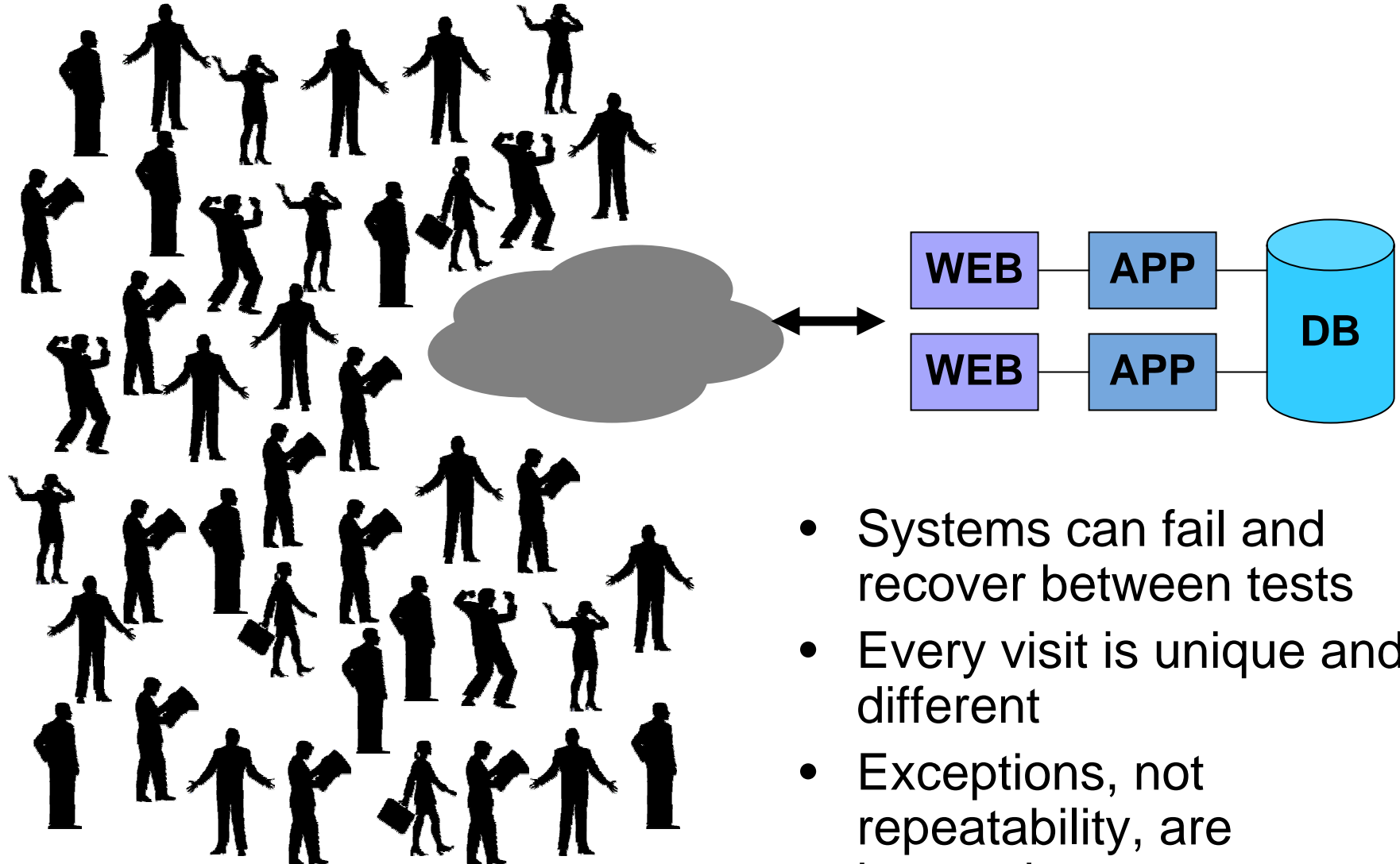
- **The business cares**
  - Real users
  - Sessions have financial impact
  - Relates load to service quality

# A service testing approach



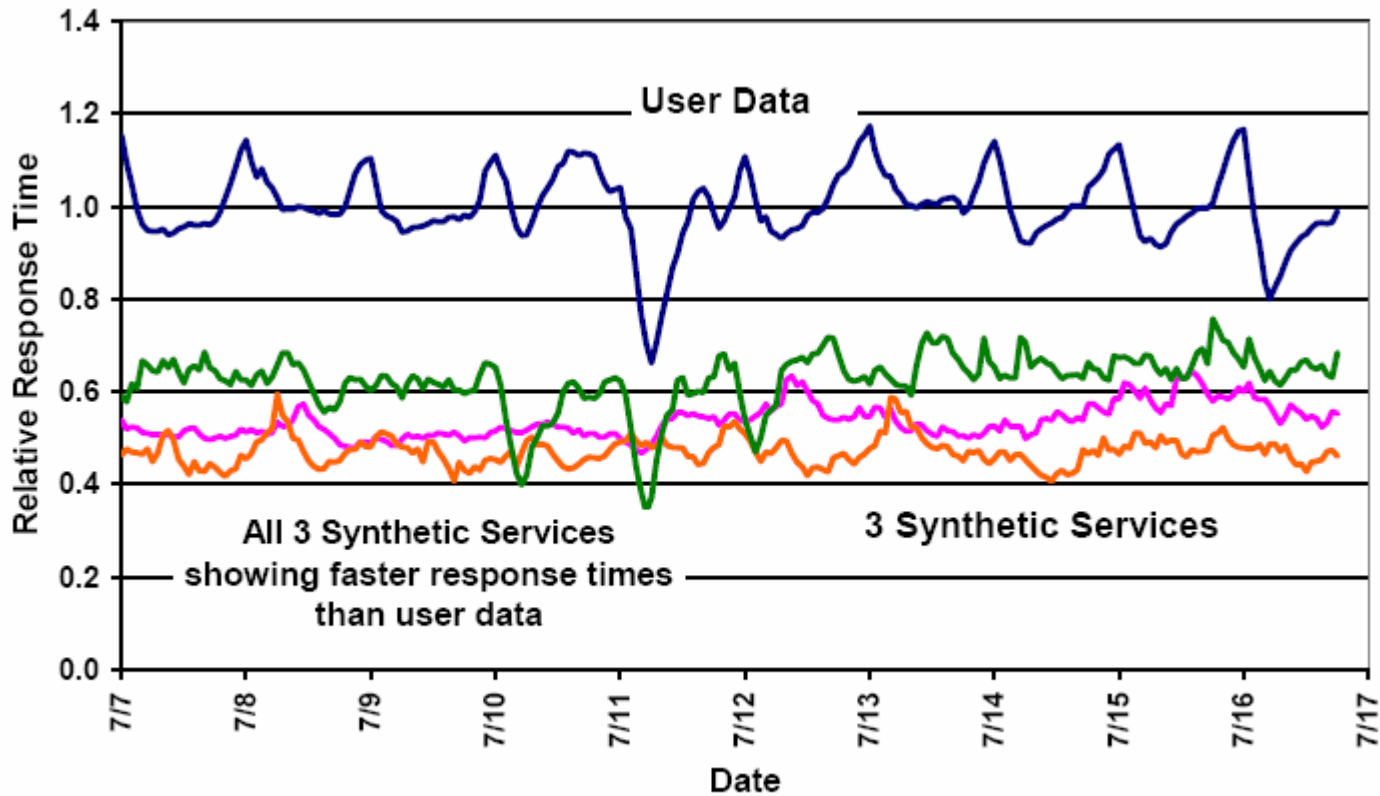
- Watch 10-15% of site
- Test specific functions you know *should* work
- From as many places as possible
- Repeatably

# Meanwhile, thousands of real sessions happen

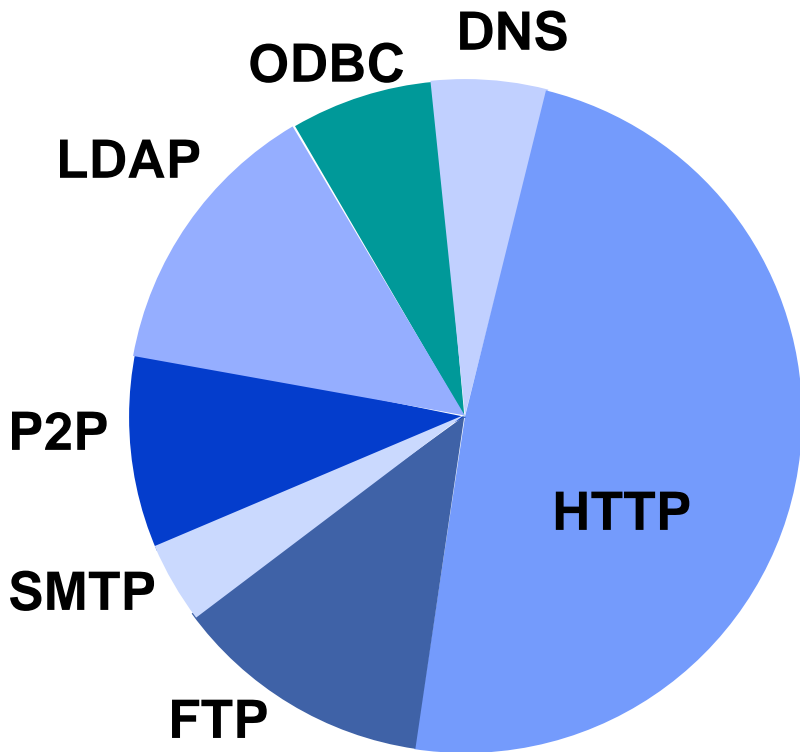


- Systems can fail and recover between tests
- Every visit is unique and different
- Exceptions, not repeatability, are interesting

# Service measurements inaccurate

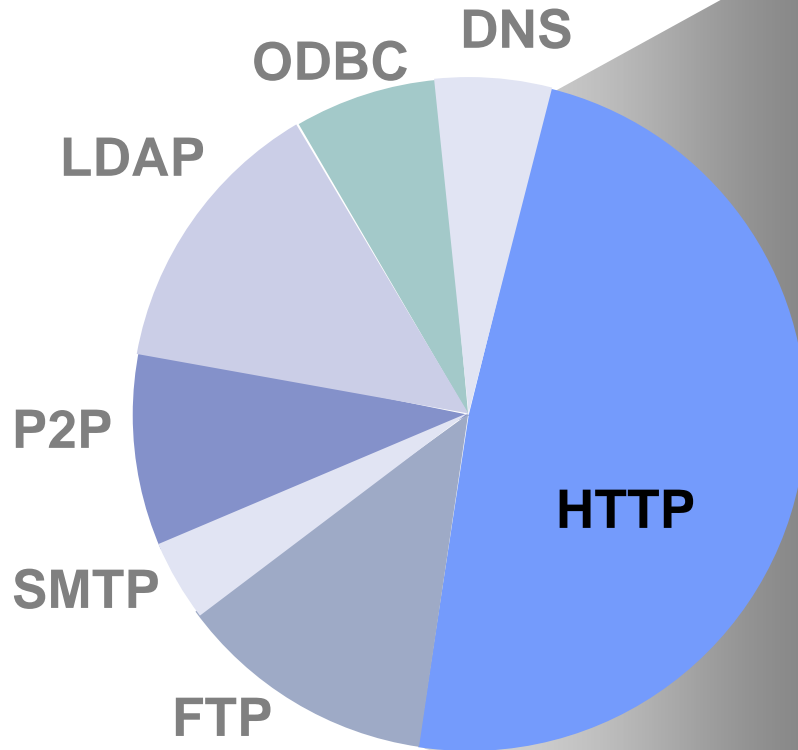


# Watching flows doesn't solve it



- Flow-based monitoring doesn't "keep up" with application state
- Shows flow details
  - Top talkers
  - Traffic by port

# But you need application context



## Client

GE	<b>Fedex</b>	GM	Dell	P&G	J&J
----	--------------	----	------	-----	-----

## Site

Finance	Mktg	<b>Legal</b>	Support
---------	------	--------------	---------

## Servers

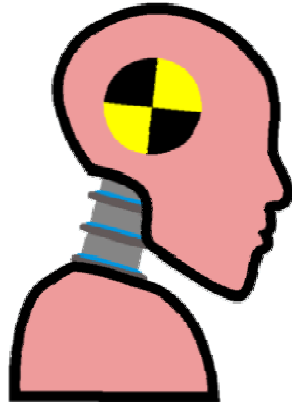
10.1	10.2	<b>10.3</b>	10.4
------	------	-------------	------

## Functions

Login	Report	Xfers	<b>Catalog</b>
-------	--------	-------	----------------

...

# Surprise: it's not either/or!



- Synthetic testing
  - Test what you know should work, repeatably, from many locations, to isolate what's beyond your control
  - Reachability, availability
  - Test key functions
  - Fake data with no privacy concerns “up to the order”

Let someone else amortize the costs, build the service



- Session monitoring
  - See what's actually happening, emphasizing the abnormal, to find problems within your sphere of control
  - Performance, load
  - Watch everything
  - Actual, confidential transactions “through the order”

Own the product, tailor to your environment and users