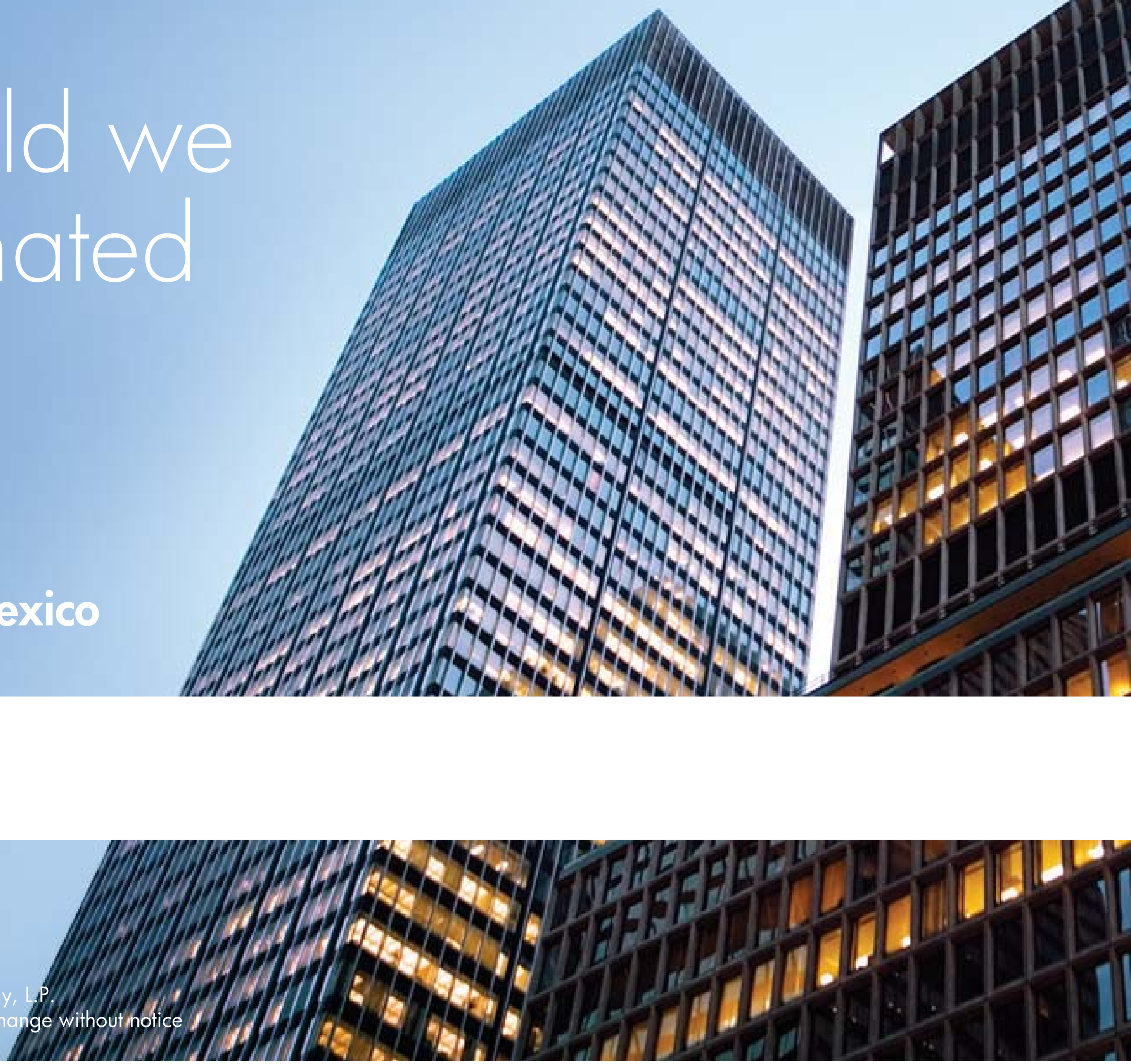
~~Why~~ How should we trust automated systems?

john wilkes, hp labs

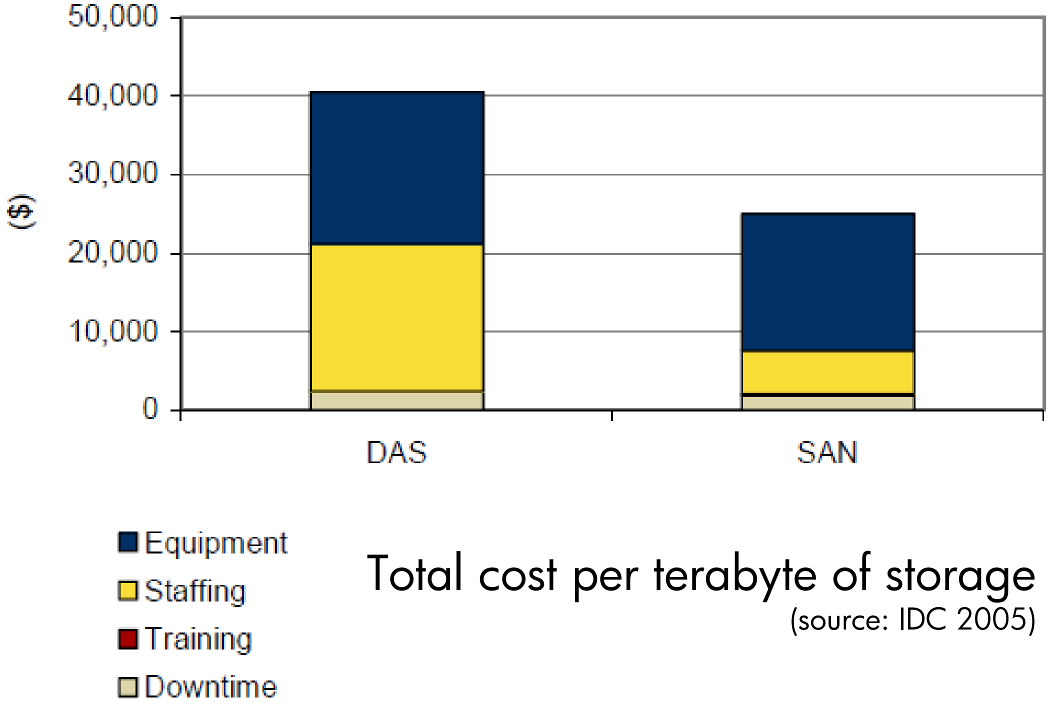
SMDB'08, Cancun, Mexico



It's inevitable
hardware vs administrator IT costs

- Storage costs are dropping
  - 1995: ~$5000/GB raw
  - 2005: $0.5/GB raw

- People costs are not:
  - 2004–5 admin salary: US$68k
  - growing ~0–6%/year [SAGE-USA survey]



Total cost per terabyte of storage
(source: IDC 2005)

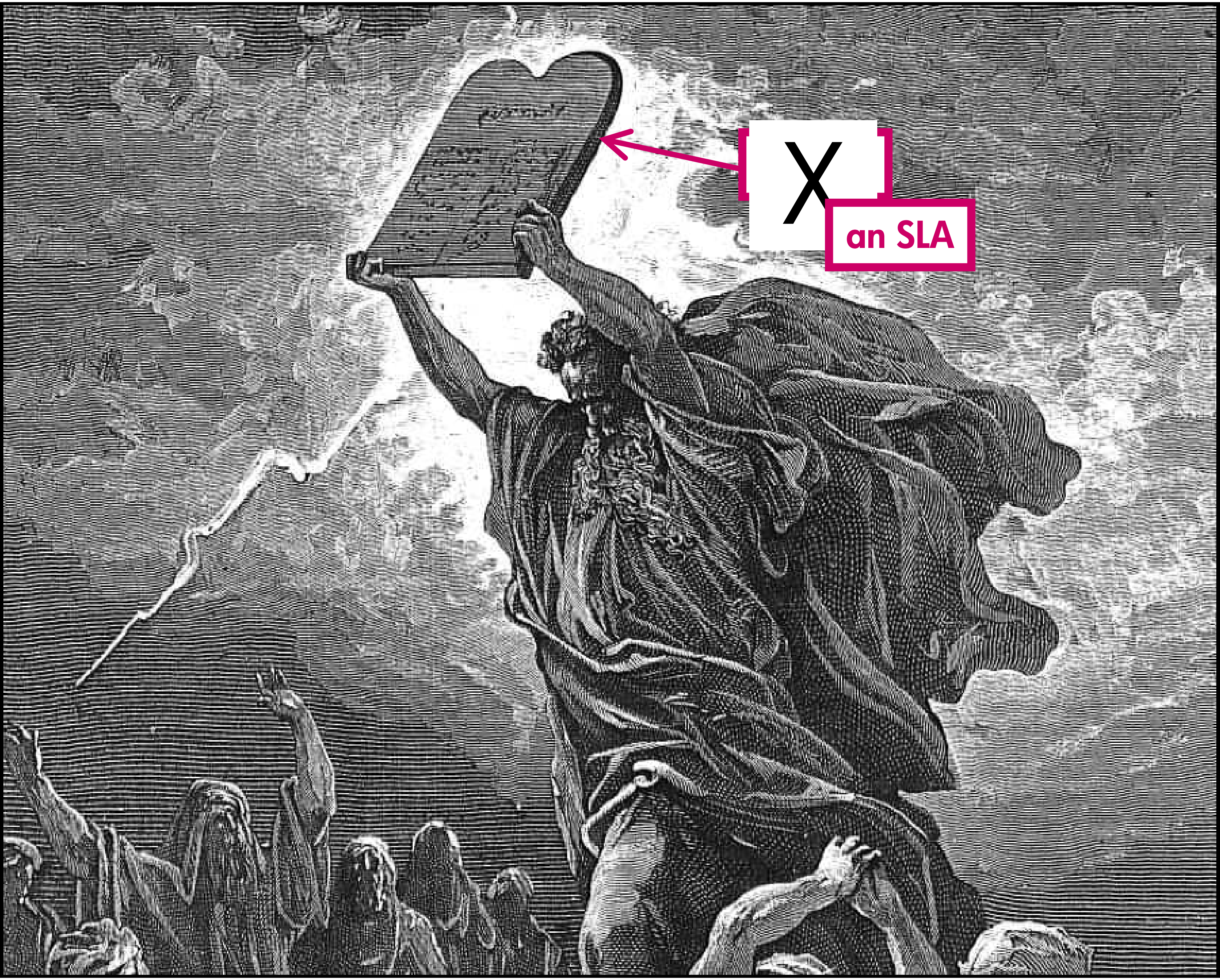
How to avoid unpleasant surprises?

- **Service Level Agreements (SLAs)**

SLAs

as contracts

- have you tried writing one?



X

an SLA

SLAs

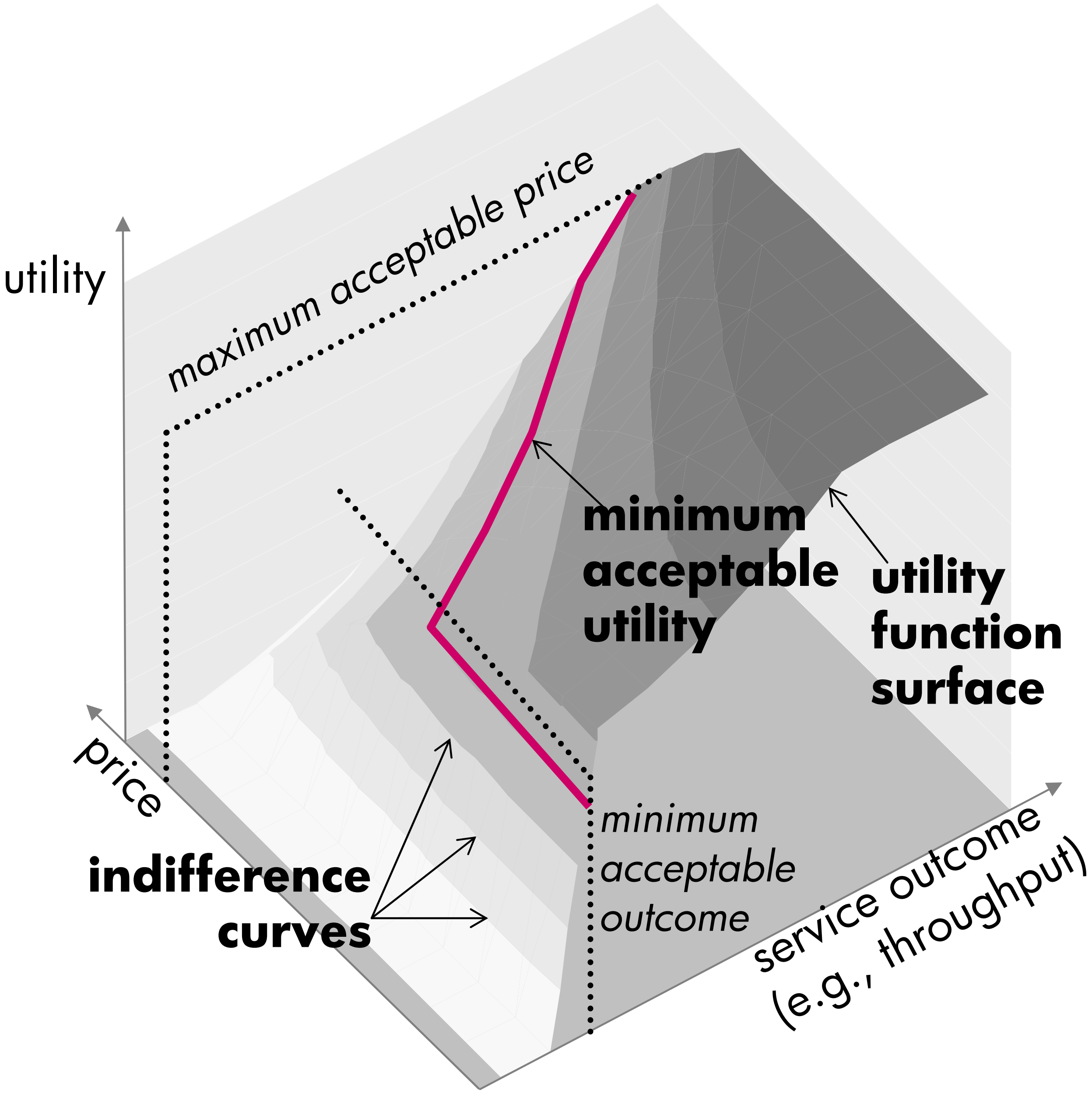
as contracts

- gospel in, garbage out?

- people are very good at coping with oddities and conflicts – computers less so
  - modal behavior (Airbus vs. Boeing)
  - rigid tradeoffs
  - ignoring "obvious" inputs

G. Doré

Doesn’t utility fix this?


utility
maximum acceptable price
minimum acceptable utility
utility function surface
price
indifference curves
minimum acceptable outcome
service outcome (e.g., throughput)

Doesn’t utility fix this?

- sure!
  - if you can extract the utility function & write it down
  - but this is hard … it’s a human data-extraction issue
  - approximations are commonplace (e.g., treat factors as orthogonal/independent – Multi-Attribute Utility Theory)

- by the way: “policies” are probably <u>not</u> the answer
  - if they mean policy rules of the form:
    if <condition> then <action>

Suggestion: treat this as a **trust issue**

- When do people accept automation?

- if they believe the *average benefits* outweigh the costs
  - e.g., "people are expensive compared to machines"

- and if they believe that the *extreme outcomes* are no worse than if mediated by a human
  - frequency
  - size of consequence

but … most people are risk averse for rare outcomes

Trust

- A **belief** that a system will "do the right thing"
  - or at least, not the wrong thing
- How established?
  - experience, more experience, and observing others' experiences (yet more experience)
  - understanding <u>why</u> outcomes are what they are
  - reassurance that the system will do the right thing

Trust

experience

- Leverage as many prior experiences as possible, not just this decision-makers'
  - reputation systems
  - explicitly presenting "similar" inputs/outcomes in response to requests
- Provide learning experiences
  - preview, then proceed
  - sure – go ahead
  - stop bugging me!

Trust

understanding why

- problem:
  - machine learning ≅ “seemed a good idea at the time”
- basic approach: explain the decisions that are made
  - expend effort on representing/visualizing the choices
  - let people drill down into proposals
  - goal: teach people to predict what the system would do

Trust

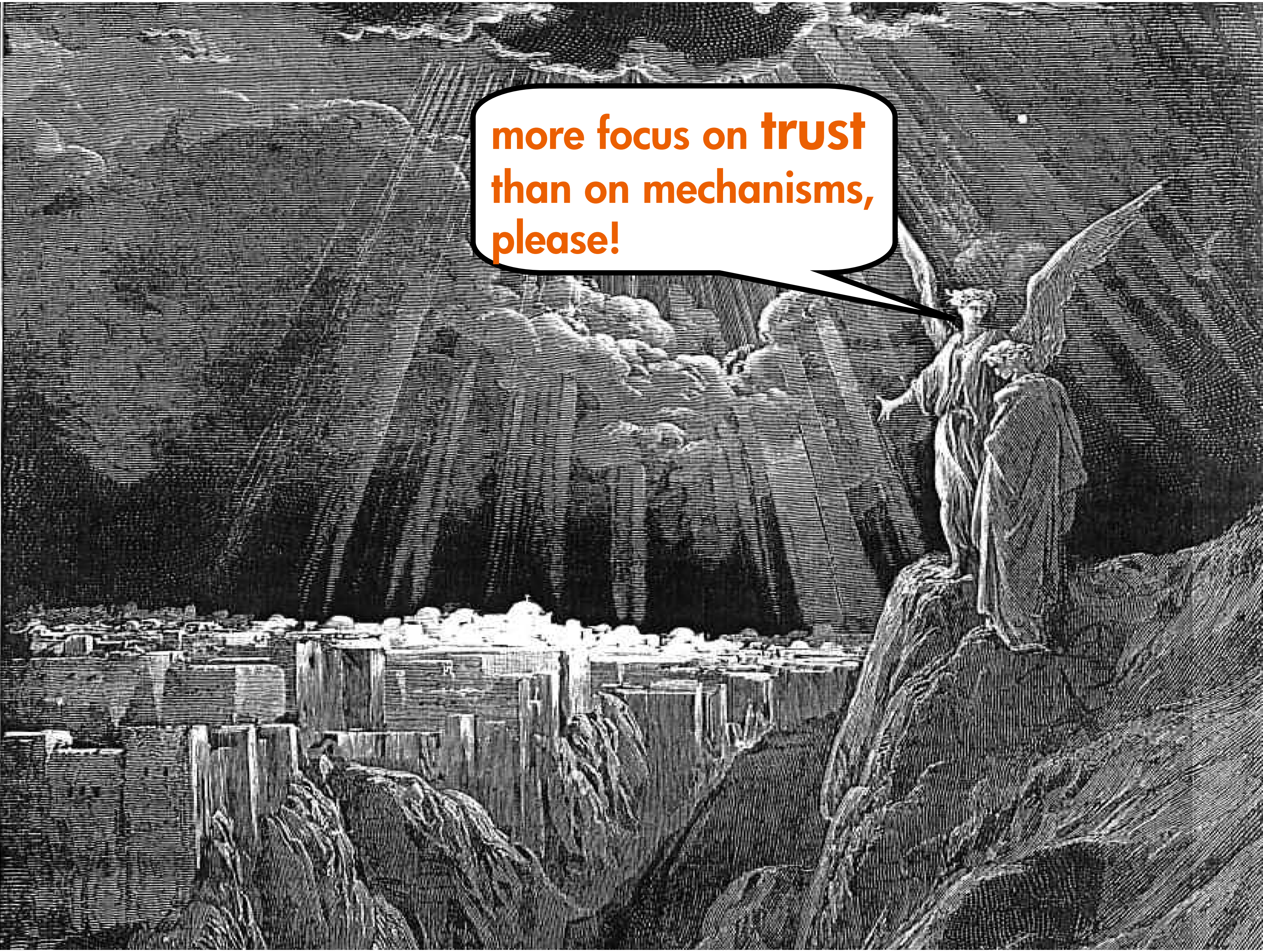
reassurance

- build in limits on outlier behavior
  - e.g., trip-wire based on size of financial consequence

  ➔ needs models of likely consequences

- auditing
  - design-time: is it likely to work?
  - deployment time: is it built + configured right?
  - runtime: is it still doing the right thing?

  ➔ need to trust the monitoring, too


more focus on trust than on mechanisms, please!

http://www.hpl.hp.com/personal/john_wilkes/papers/#Tuscany