Microsoft

# Microsoft Research Faculty Summit 2013

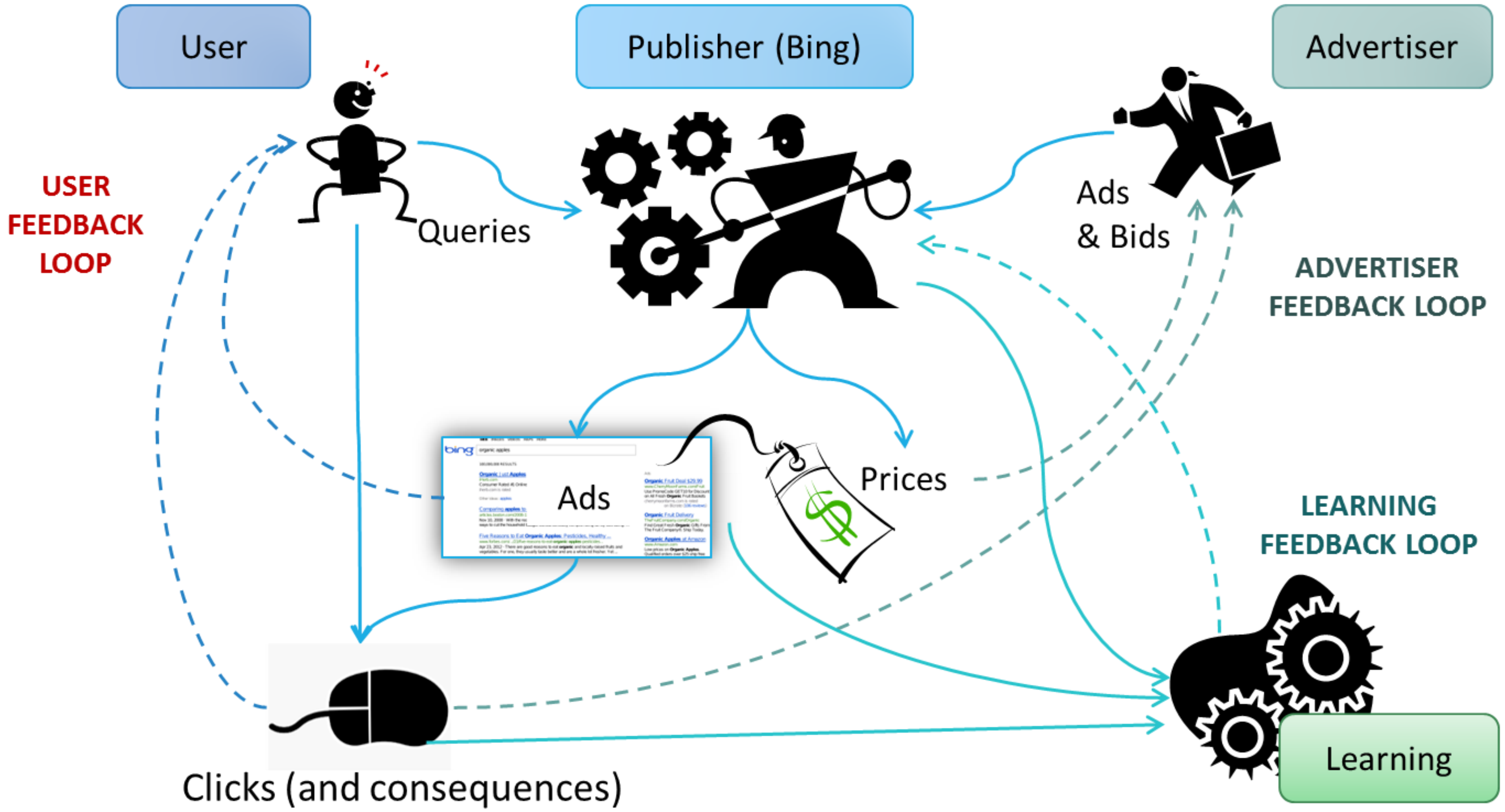
Microsoft

# Interactive learning systems: Lessons learned from ad placement.

Leon Bottou

Microsoft Research
Faculty Summit

# Ad placement is difficult



# Issues

## Main relevant scientific topics

- Auction theory
  (mechanism design, placement auctions, …)
- Learning with limited feedback
  (sequential design, explore/exploit, bandits, …)

## Engineering issues

- Team work + big data

# Applying scientific insights

## Sound scientific approach

- Focus on the simplest setup that exhibits the phenomenon of interest and is amenable to analysis

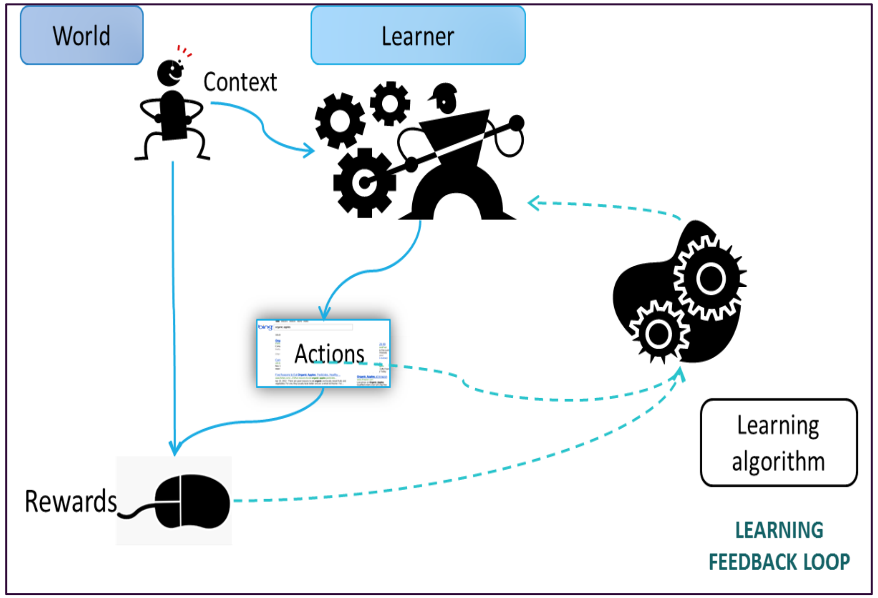
## Practical consequences

- Setup too restrictive to apply
- Setup too general to lead to competitive system
- Both of the above

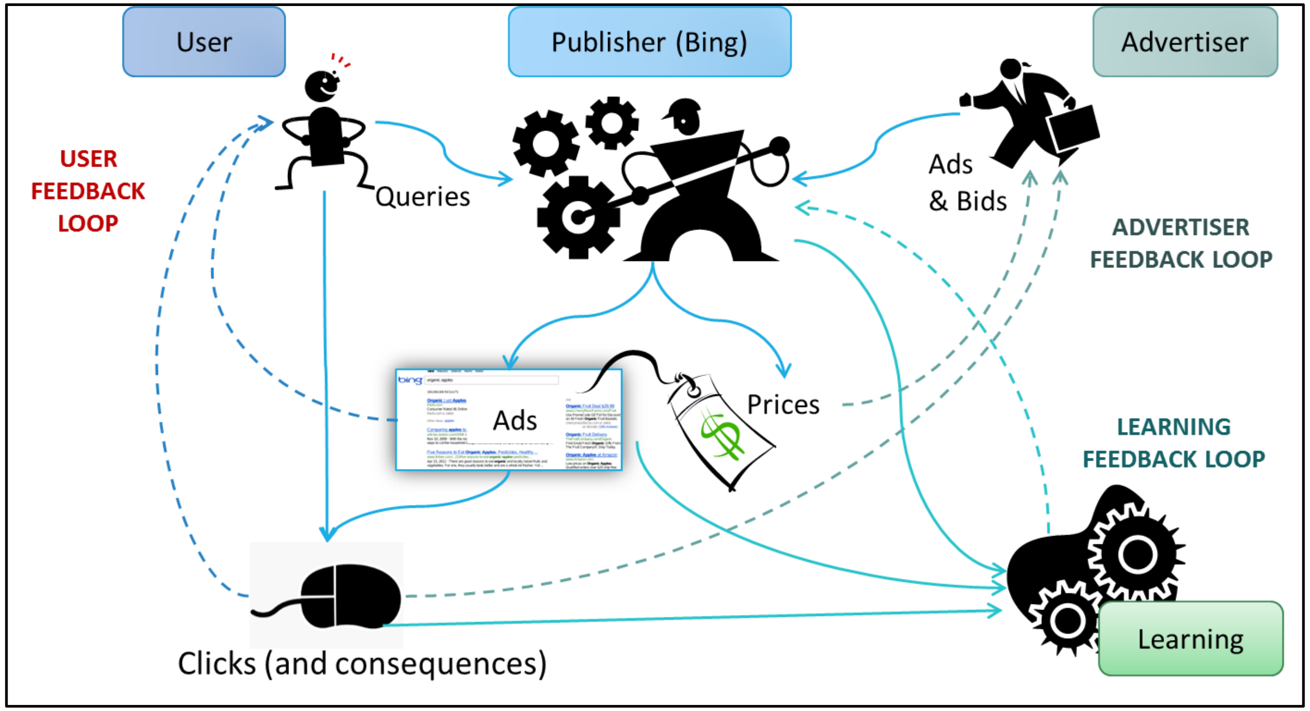
# Auction theory for ad placement?

- Many queries are targeted by a single advertiser.
  - When there is only one buyer, this is not an auction!
- Optimal auction theory does not (usually) apply to repeated auction.
  - Repeated business gives more leverage to the buyers
- Advertisers place a single bid for multiple auctions.
  - Ad placement engines serve hundreds of millions of queries per day.
    The most active advertisers change their bids every 15 minutes.
- Placement decisions impact the future behavior of users.
  - Auction theory models the interaction of one seller and many potential buyers.
    publisher = seller, advertisers = buyers, user = ?

# Contextual bandits for ad placement?

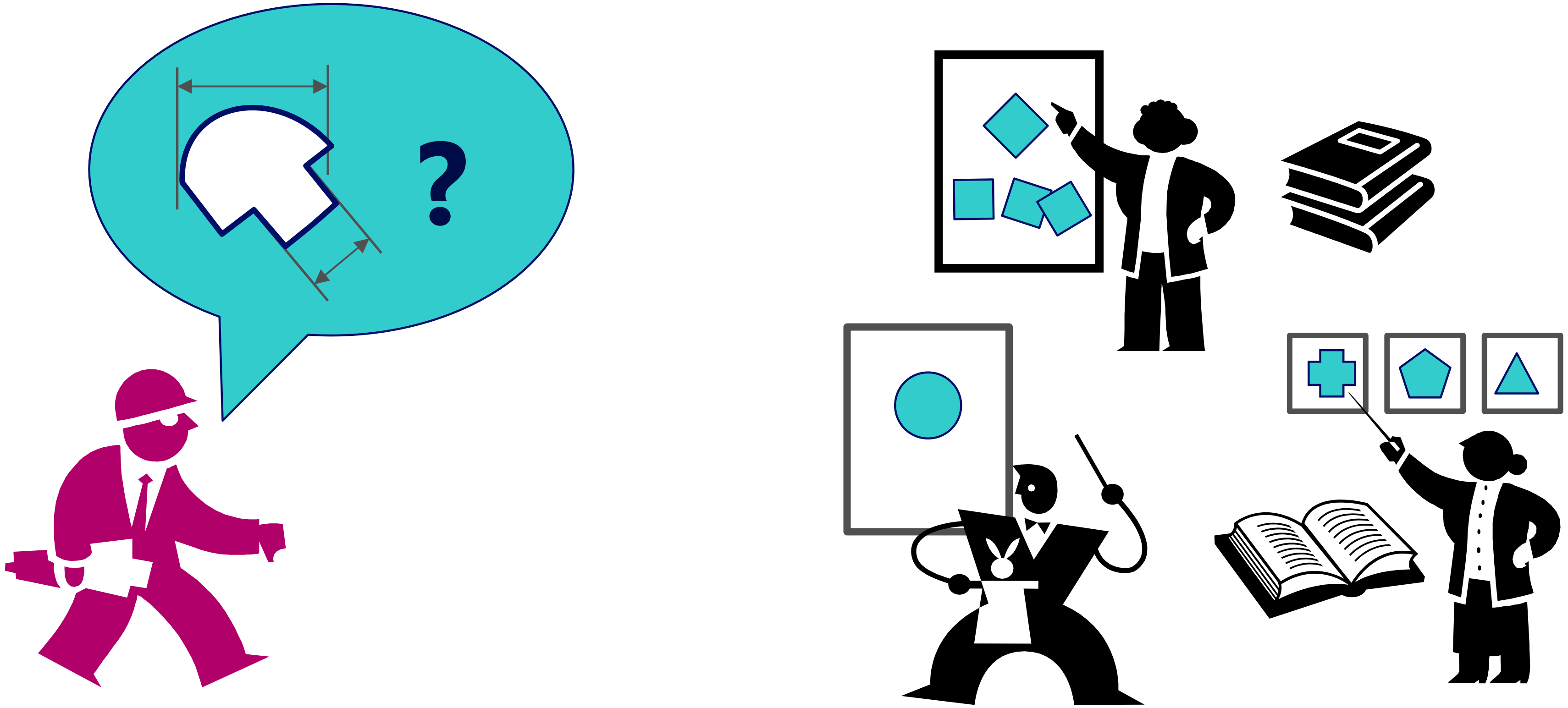



≠




- Missing **user feedback loop**, missing **advertiser feedback loop.**
- Does not exploit **action structure** (similar ads on similar queries), **policy structure** (ad auctions must obey certain rules) or **reward structure** (pricing decisions affect users but not advertisers.)

# How to help the engineer?



# Listening to the question

**Narrative**

- Collected data shows a positive correlation between conditions A (e.g., some ad feature) and B (e.g., clicks),
- But when we change the ad placement engine to get more A we do not get more B.

**Questions**

- What is going on here?
- Why do such things happen all the time ?
- How can I engineer such a system?

# Reichenbach's common cause principle

A and B are correlated ⇒

- A causes B
- or B causes A
- or A and B have a common cause C.

# Reichenbach’s common cause principle

A and B are correlated ⇒

- ~~A causes B~~
- or ~~B causes A~~
- or A and B have common causes.

Were this the case, manipulating A would change B as expected.

Impossible because B follows A in time.

By elimination

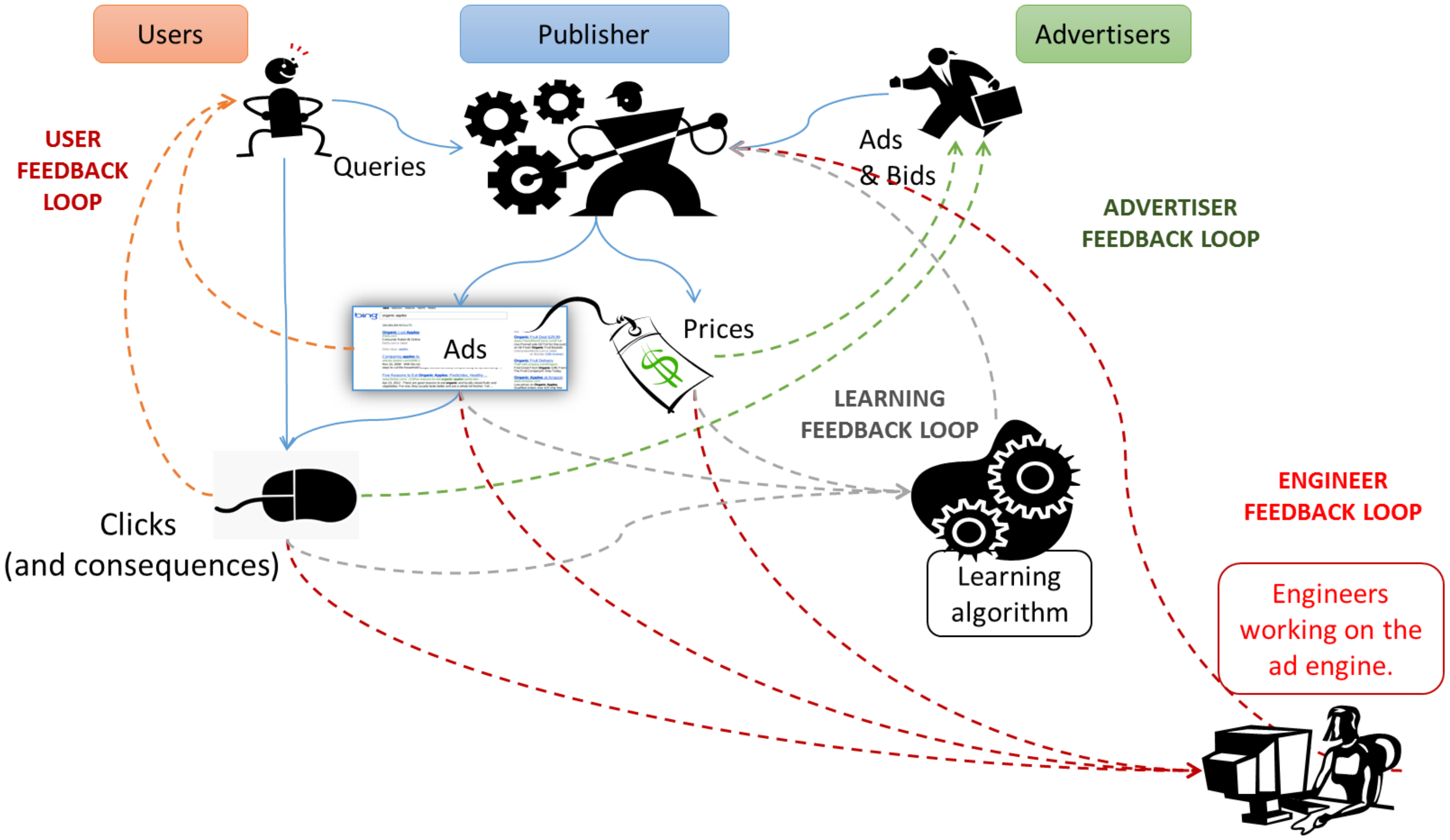
# Reichenbach's common cause principle

Manipulating A should not be expected to change B as the correlation suggests !

A and B are correlated $\Rightarrow$

- ~~A causes B~~
- or ~~B causes A~~
- or A and B have common causes.

# Humans are part of the learning system



# Revisiting the question

Give us a **framework** to reason about such problems.

- A **generic language** should we use to express the assumptions that **we believe adequate** for the problem,
- with **generic methods** to **construct sound learning algorithms** tailored to our assumptions.
- and **generic methods** to **construct sound monitoring techniques** to validate assumptions, check the learning process at any time, debug problems, etc.

# Solving the framework problem

Write a collection of papers with incompatible setups illustrating relevant insights.

→

Express insights within a unified framework that provides a generic modeling language and generic methods.

- My bets are on causal inference.