Egg: An Extensible and Economics-Inspired Open Grid Computing Platform

David C. Parkes

Division of Engineering and Applied Sciences
Harvard University
GECON 2006

Grids

- internal realm:
  - Java, python, C++, applications
  - Science, engineering, art
  - Happy
- external realm:
  - OS. Disk. WAN. Firewall. HTTP. Installation.
  - People, organizations
  - Labor intensive. (54 sites for OSG)
  - Sad

Current (Science) Grids

- No global resource allocation mechanism
- Installing and maintaining grid infrastructure software is time-intensive and difficult
- Converting applications to be grid-enabled is time-intensive and difficult
- Complex to express user and organizational policies, user needs

What is Egg?

- Egg == Extensible and Economics-Inspired Open Grid Computing Platform

- Goals: open, efficient, simple grid computing, respect organization boundaries
- "Programming the external world"
- Collaboration: CS + Physics + Economics
  - Boston University, Harvard University
  - L.Kang, C.Ng, M.Seltzer, D.Parkes
  - J.Brunelle, P.Hurst, J.Huth, J.Shank, S.Youssef
  - A.Sunderam

In the beginning…

Boston University	Harvard
Software environment computing, i.e. creating and manipulating software environments	Economic mechanism design; bidding systems, provenance & file systems, resource prediction

+ Collaboration on ATLAS, several years of experience with Globus-based Grids and BU's new ATLAS "Tier 2" center.

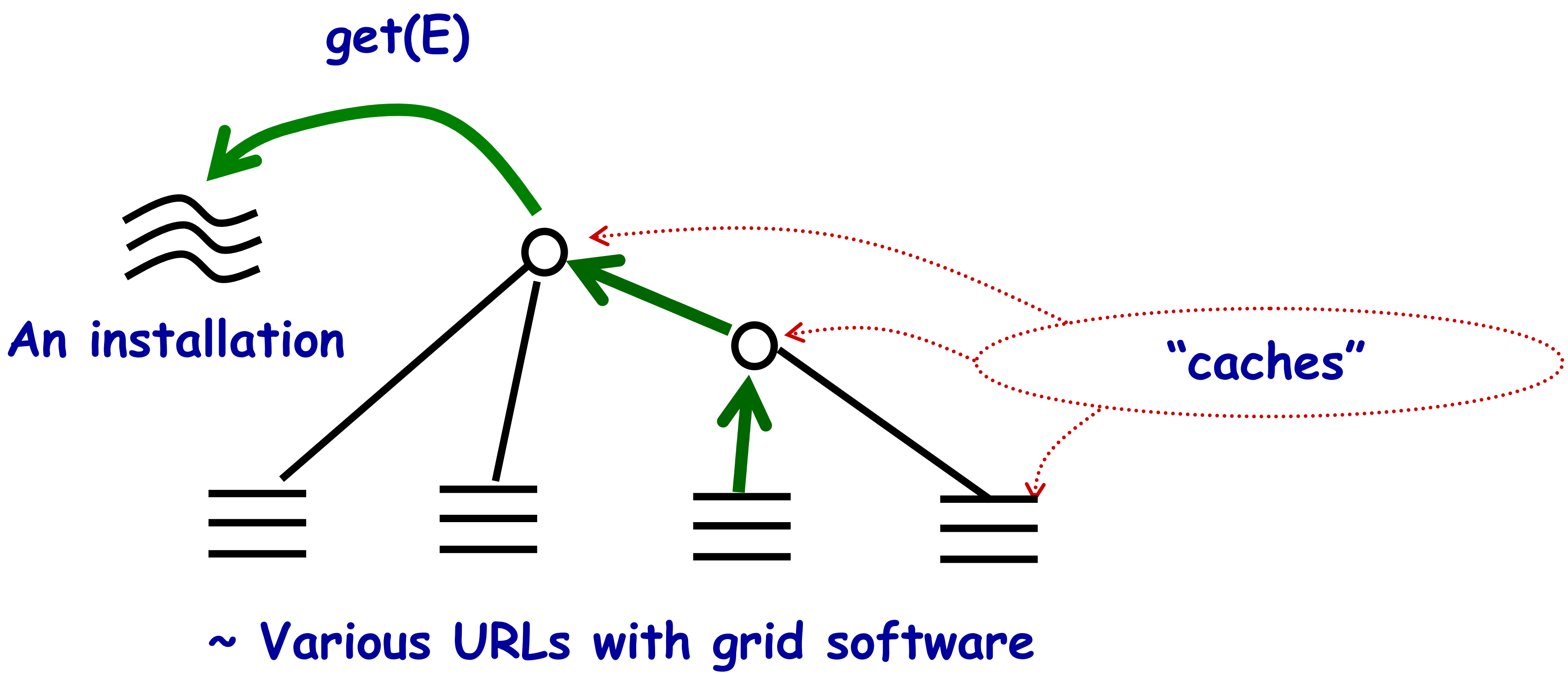
In the beginning…

Boston University	Harvard
Software environment computing, i.e. creating and manipulating software environments	Economic mechanism design; bidding systems, provenance & file systems, resource prediction

But what do these have to do with each other? …And how do they fit into the (over-)complicated world of grid computing?

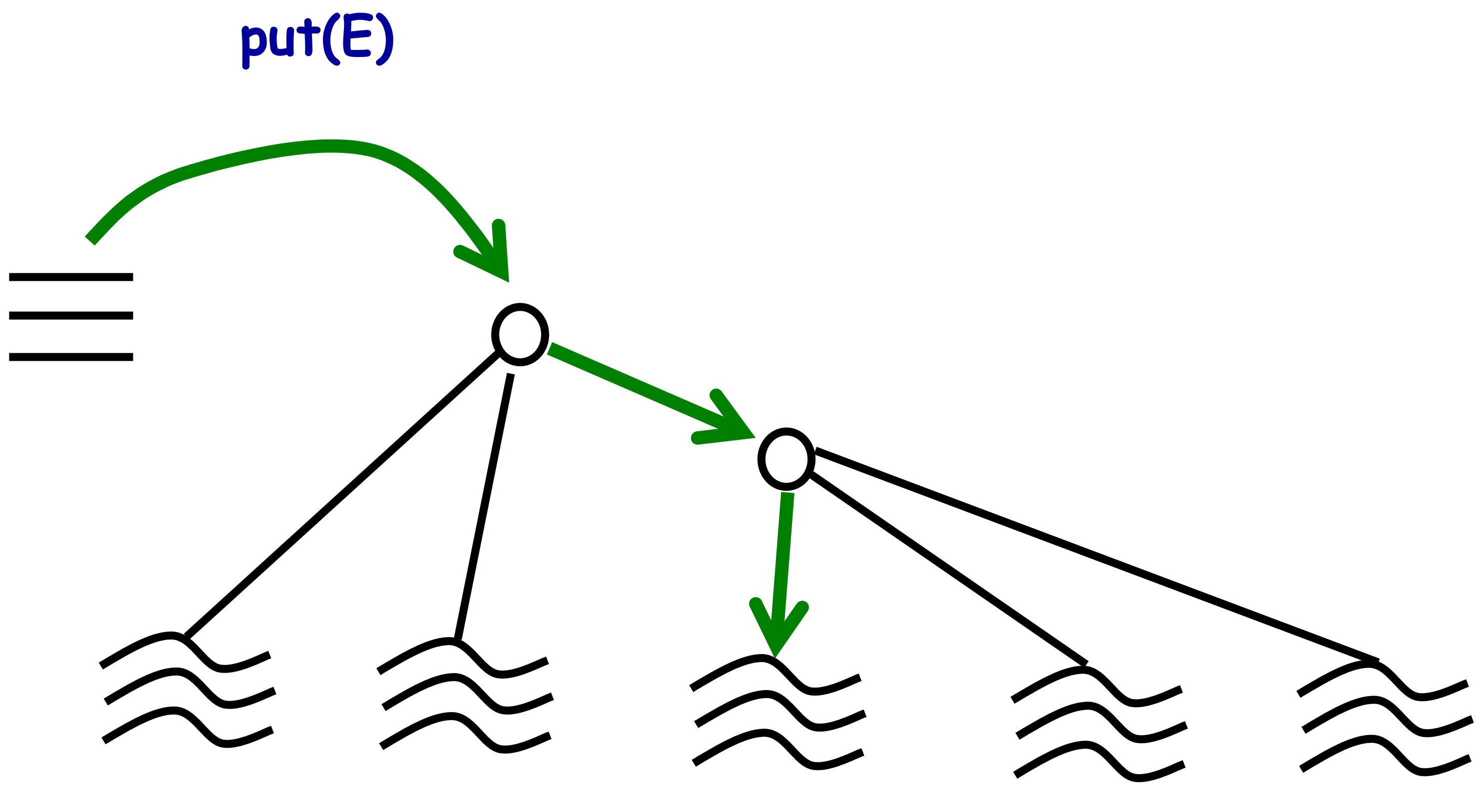
Netlogger
dCache
Condor
Alien
Ganglia
Panda
VDT
GLOBUS
Pacman
Resource Brokers
PBS
Chimera
SRM
iVDGL
ADA
Gums
Web services
VDS
LSF
RLS
EGEE
Dirac
Capone
VOMS
OSG
Eowyn
Dial
Glue
gLite
PPDG
MonaLisa
Virtual Machines
EDG
Clarens
LCG
GridCat
DISUN
DRM
ACDC
Classads

To begin, let's think about "Pacman" (S.Youssef, BU)



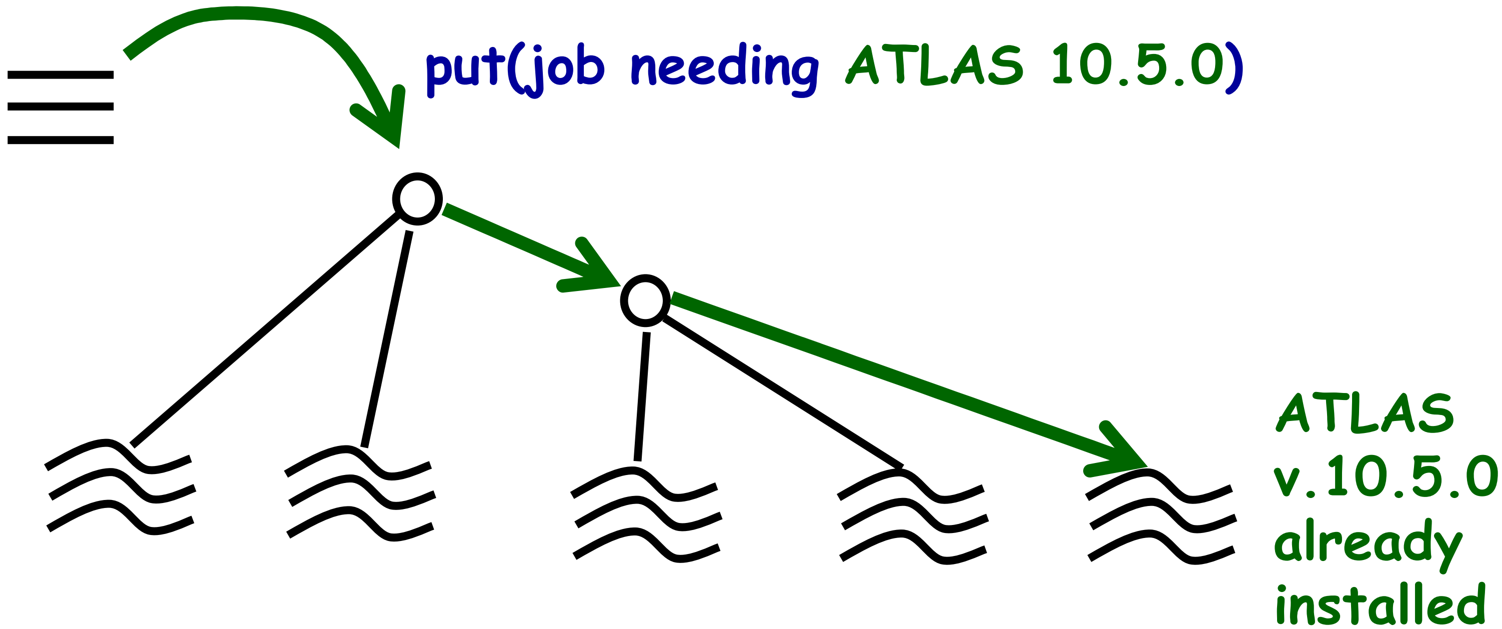
[Pacman is used by ATLAS (>1800 physicists, >150 labs, 34 countries), OSG, Virtual Data Kit (incl. Condor and Globus), TeraGrid,… >800,000 Pacman downloads (3/12/06), ~1000 new installations per day in 50+ countries, supported on 14 OS.]

We can let all computations be "installations."



But which path should E follow?

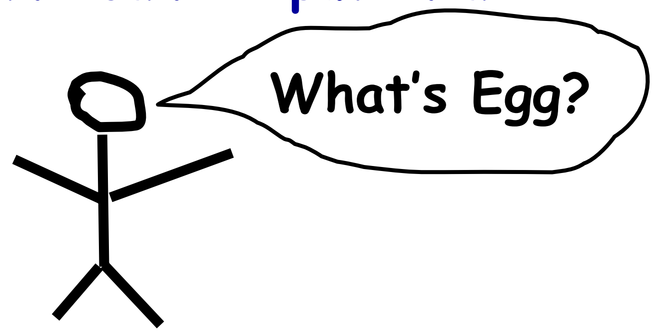
Resolving the put ambiguity == Resource allocation



F(Job description , Cache history , Cache contents)

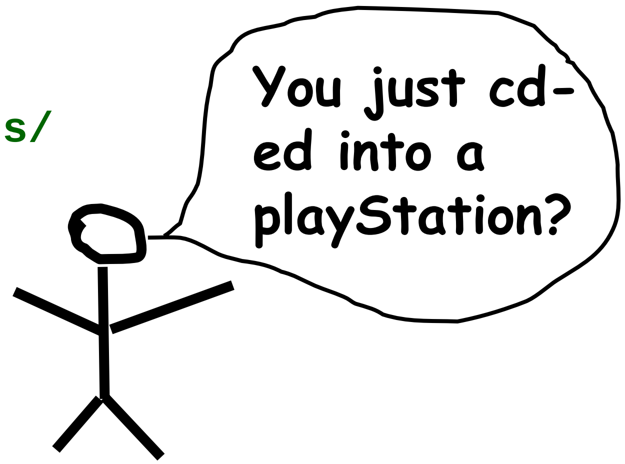
⇒ ~Opportunity cost

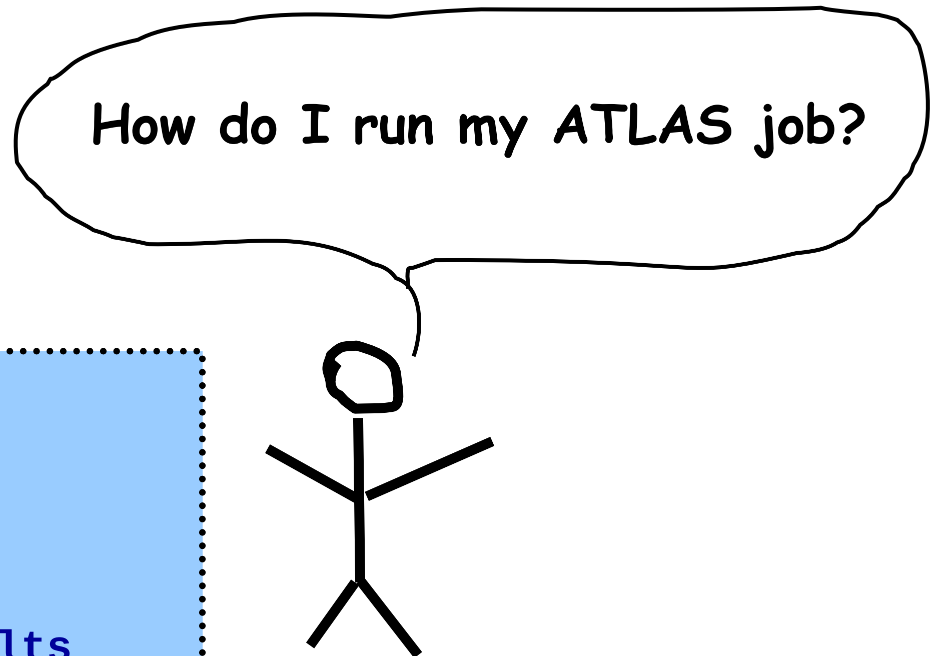
User level concepts can be simple and can be put in a familiar, easy to learn context.




```
% egg

egg> cd ~David

egg> lc

myEgg.caches hu.playCluster david.grid david.playStation
results/ papers/ jobs/ Tier2/ identities/

egg> cd jobs

egg> lc

job1.eggshell job2.eggshell job3.eggshell

egg> put job2.eggshell ../david.playStation

egg> cd ../david.playStation

egg> lc

queue/ running/ history/ earnings/ access/
```

```
egg> lc
queue/ running/ history/ earnings/ access/
egg> lc -r
queue/
    job2.eggshell
         ATLAS.Higgs.HU.David:10@
running/
    job1.eggshell
         ATLAS.Higgs.HU.David:10@
         results/
             seeds higgs.aod athena.log error.log
earnings/
    ATLAS.Higgs.BU.Saul.CANCELLED:10@
    Harvard.EECS.Margo.Laura.CANCELLED:1@
access/
    *.Saul
    *.Margo.?
```





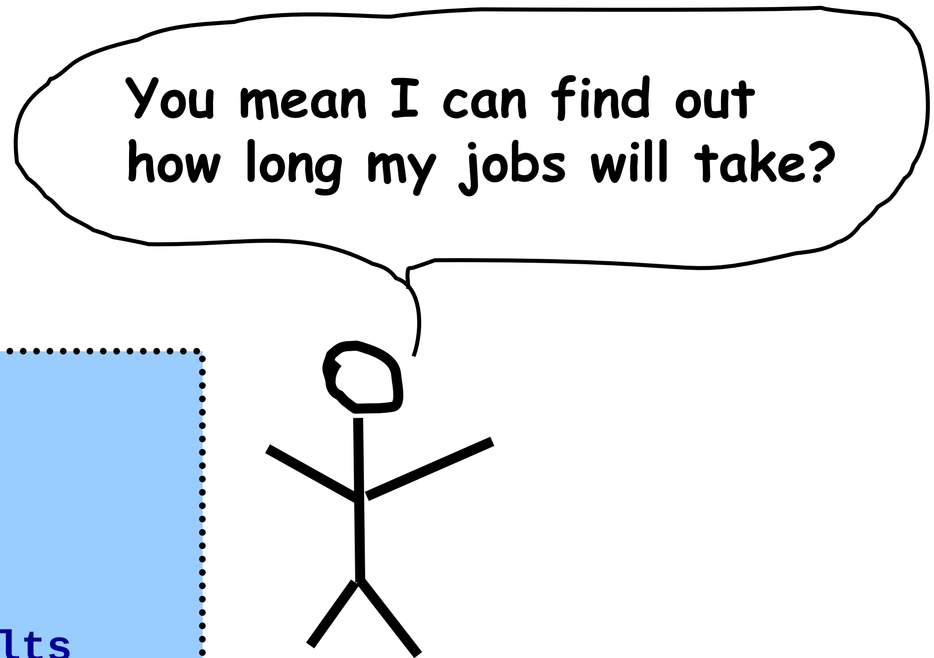
job1.eggshell

```
put ~ATLAS/10.5.0 .

put ~David/jobs/binary1 .

put ~David/jobs/job1.in .

put ./results/job1.out ~David/results

pay ATLAS.Higgs.HU.David:10@
        when gmTime < 1-Apr-2006

shell echo “done”
```

```
egg> put job1.eggshell ~David/mygrid

egg>
```

job1.eggshell

```
put ~ATLAS/10.5.0 .

put ~David/jobs/binary1 .

put ~David/jobs/job1.in .

put ./results/job1.out ~David/results

pay ATLAS.Higgs.HU.David:10@
       when gmTime < 1-Apr-2006

shell echo “done”
```




```
egg> put job1.eggshell ~David/mygrid

egg> lc ~David/mygrid/job1*

job1.eggshell

       e.t.a. 25-Mar-2006 +- 2 days

       estimated cost ATLAS:Higgs.HU.David:8.3@
```

Main Innovations

- Microeconomics. All actions (installations, downloads, uploads, etc.) are put and gets. Made efficient by bidding mechanism. Simple + transparent to users.

- Macroeconomics. Multiple currencies. Policy autonomy. Support for interoperation between grids. Simple + transparent to users.

- Open + Extensible. E.g., IBM can develop own bidding agent for its compute servers.

Main Innovations

- Microeconomics. All actions (installations, downloads, uploads, etc.) are put and gets. Made efficient by bidding mechanism. Simple + transparent to users.
  - reverse auctions, role of Egg platform
- Macroeconomics. Multiple currencies. Policy autonomy. Support for interoperation between grids. Simple + transparent to users.
  - currency, exchange rate, role of banks
- Open + Extensible. E.g., IBM can develop own bidding agent for its compute servers.

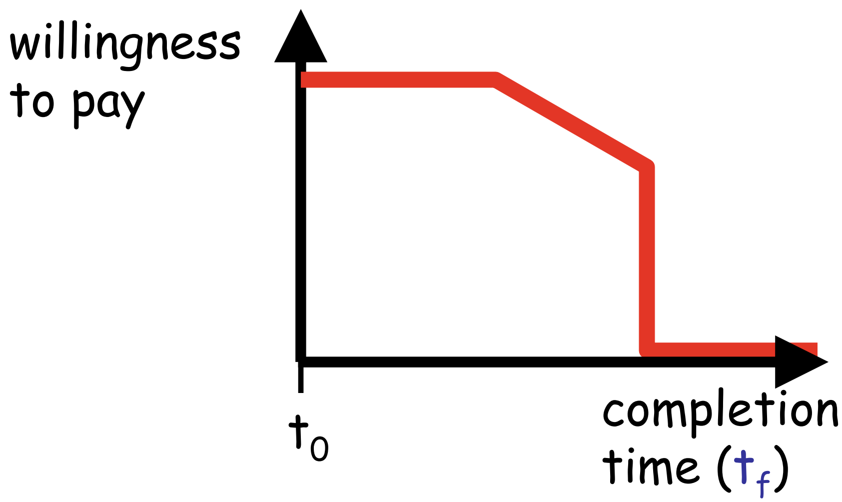
Novelty: Open mechanism design

- Open: unrealistic to propose a particular selling mechanism that all resource owners should use
- Dynamic, distributed, asynchronous
  - e.g., a single, centralized, forward combinatorial auction would not work

- **Our solution: Egg platform places constraints** on mechanisms (price tables, admissibility)

First: User Expressiveness

- Describe Job in Eggshell
  - executable files, input files, loops, etc.
  - maps to bundle S of resources
- Describe a "value schedule" $v_i(S, t_f)$.



- Simplify for users via default schedules

Now to Open MD: Price Admissibility

admissible prices == user i faces a price, $p^t_i(S,t_f)$, in period t, for bundle S and completion by t_f that is:

(a) independent of agent i

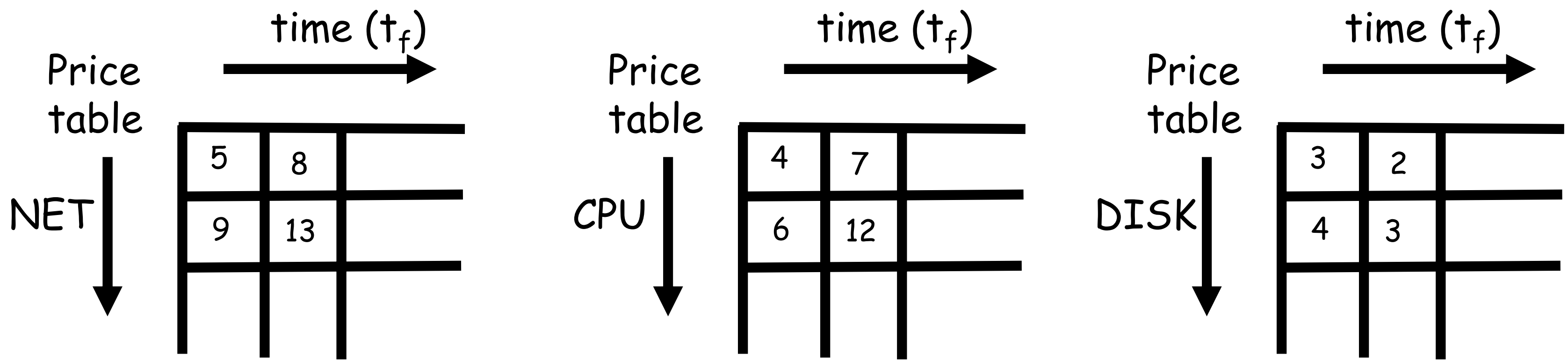
(b) increases monotonically with $S' \supset S$

(c) increases monotonically with current time, t

➡ A reverse auction with admissible prices, and in which agent i receives completion time t_f that maximizes $v_i(S,t_f)-p^t_i(S,t_f)$, is strategyproof.

$\Rightarrow$ Egg enforces monotonicity of prices wrt **S** and **t** through price tables; enforces maximal decision.

Price Tables

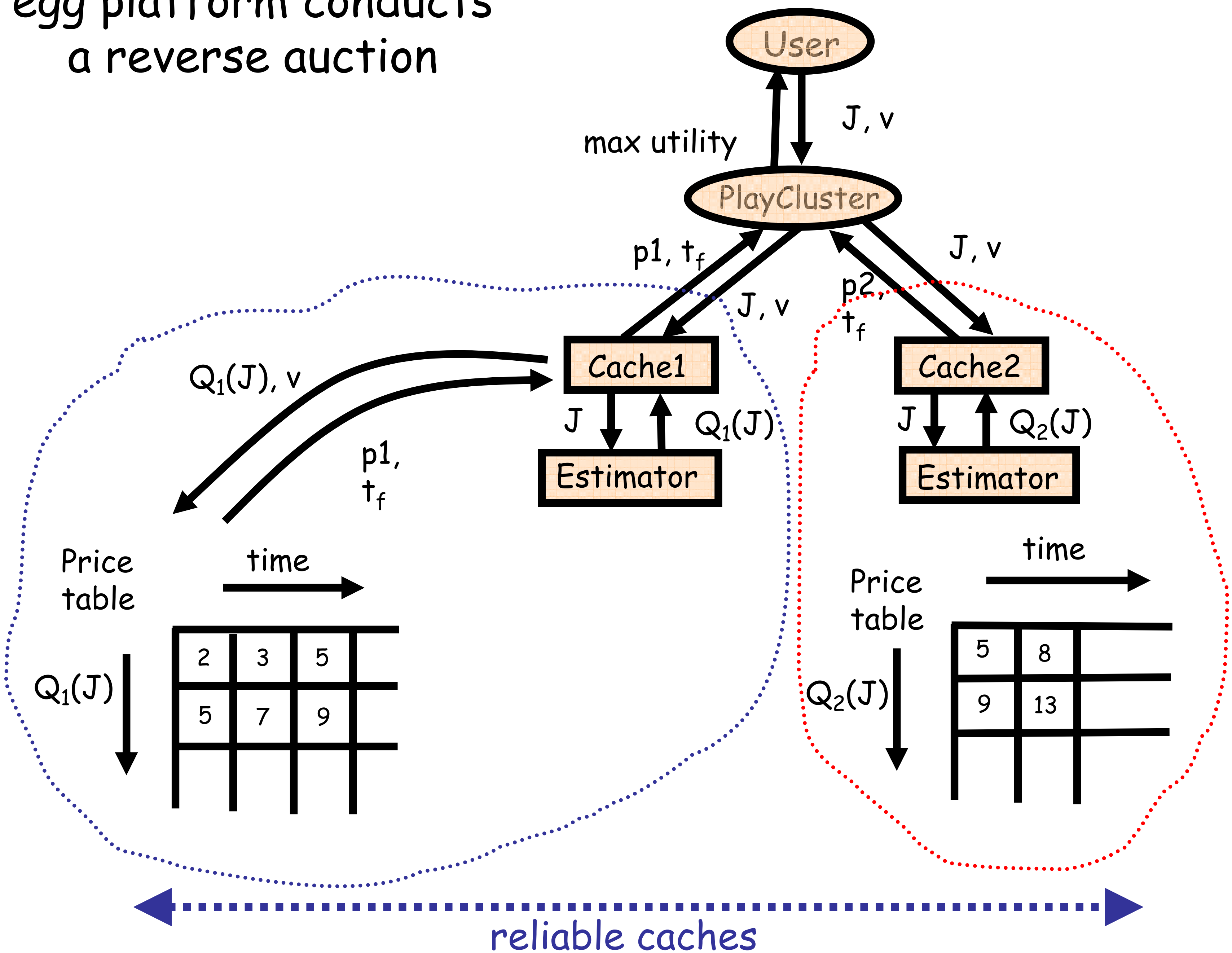


$$p^t_i(S,t_f)=p_i^{NET}(S_{net},t_f)+p_i^{CPU}(S_{cpu},t_f)+p_i^{DISK}(S_{disk},t_f)$$

Caches maintain entries in price tables (but, cannot reduce prices, & must retain monotonicity w/ size.)

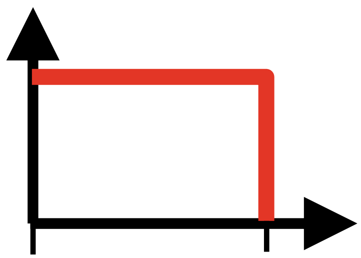
egg platform enforces this

egg platform conducts a reverse auction



Example: Buying Storage

- “Deadline 5hrs”, estimated space is 2GB for 2 hrs.



Cache’s price table

time (hrs) →

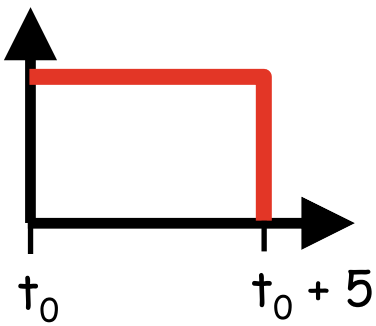
Disk space ↓

	t_0+ 1	+2	+3	+4	+5		
1G	4	3	5	8	6		
2G	5	8	6	9	7		
3G	6	9	8	12	9		

Collate responses. Choose to allocate to best cache.
Only pay if completed by estimated time.

Example: Buying Storage

- "Deadline 5hrs", estimated space is 2GB for 2 hrs.



Cache's price table

time (hrs) →

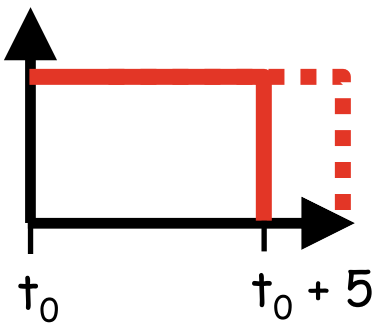
Disk space ↓

Disk space	t_0+ 1	+2	+3	+4	+5		
1G	4	3	5	8	6		
2G	5	8	6	9	7		
3G	6	9	8	12	9		

Should not be $3 (monotonicity)

Example: Buying Storage

- "Deadline 5hrs", estimated space is 2GB for 2 hrs.



Cache's price table

time (hrs) →

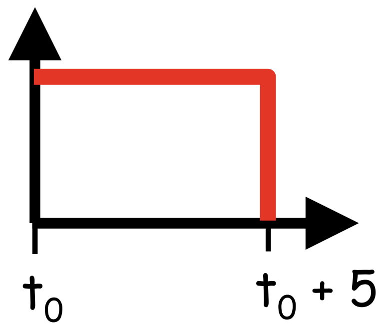
Disk space ↓

	t_0+ 1	+2	+3	+4	+5	+6	
1G	4	3	5	8	6		
2G	5	8	6	9	7	**2**	
3G	6	9	8	12	9		

Suppose (2G,6) is $2. Better to over-report deadline?

Example: Buying Storage

- "Deadline 5hrs", estimated space is 2GB for 2 hrs.



Cache's price table

time (hrs) →

Disk space ↓

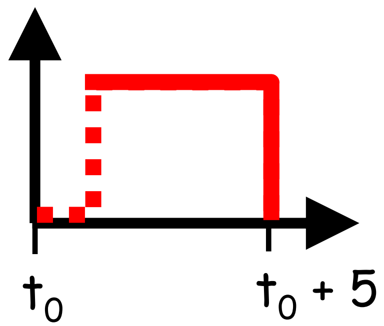
		t_0+ 1	+2	+3	+4		
1G	4	3	5	8	6		
2G	5	8	6	~~9~~	7		
3G	6	9	8	12	9		

4

Suppose time ticks forward, and price in (2G,+3) falls.

Example: Buying Storage

- “Deadline 5hrs”, estimated space is 2GB for 2 hrs.



Cache’s price table

time (hrs) →

Disk space ↓

		t_0+ 1	+2	+3	+4		
1G	4	3	5	8	6		
2G	5	8	6	~~9~~	7		
3G	6	9	8	12	9		

4

Delay “arrival.” Payment $10, not $13.

Next steps: Micro

- **Resource estimation via machine learning**
  - Statistical learning problem
  - Learn $g : job \rightarrow R^k$
    - for k dimensions of local resources
  - Each cache keeps local history
  - Updates model (g)
  - Consider linear-regression trees, SVMs, k-nearest neighbor…
- **Bidding strategy by caches**
  - Decision theoretic problem
  - Maximize expected revenue subject to capacity constraints, price-table monotonicity constraints
  - Consider model-based approach, w/ estimate of success for different prices

Main Innovations

- Microeconomics. All actions (installations, downloads, uploads, etc.) are put and gets. Made efficient by bidding mechanism. Simple + transparent to users.
  - reverse auctions, role of Egg platform
- Macroeconomics. Multiple currencies. Policy autonomy. Support for interoperation between grids. Simple + transparent to users.
  - currency, exchange rate, role of banks
- Open + Extensible. E.g., IBM can develop own bidding agent for its compute servers.

Need for Policy Tools

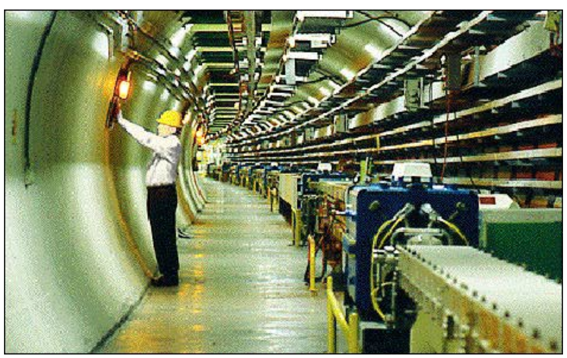
- Large-scale collaborative science is often driven by organizations.
- Organizations want high-level control over resource access, priorities, etc.
- Policy requires control of currency [to print, allocate, tax, etc.]

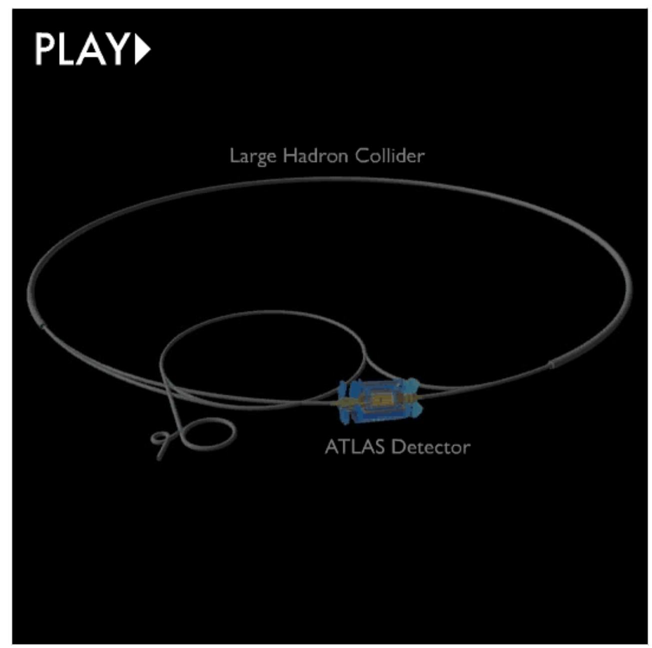
  $\Rightarrow$ need multiple currencies

  (or tight integration, c.f. EU)

Motivating Scenario I: ATLAS project

- Create ATLAS bank and ATLAS eggs. Create accounts for research scientists. (Tier 2 at BU. Jim Shank comput. manager for ATLAS)
- Meeting of stakeholders: searching for Higgs boson over next month requires 70% global resources
- Print and allocate ATLAS currency to Higgs manager, who further delegates. Users retain control about which analysis jobs to run.



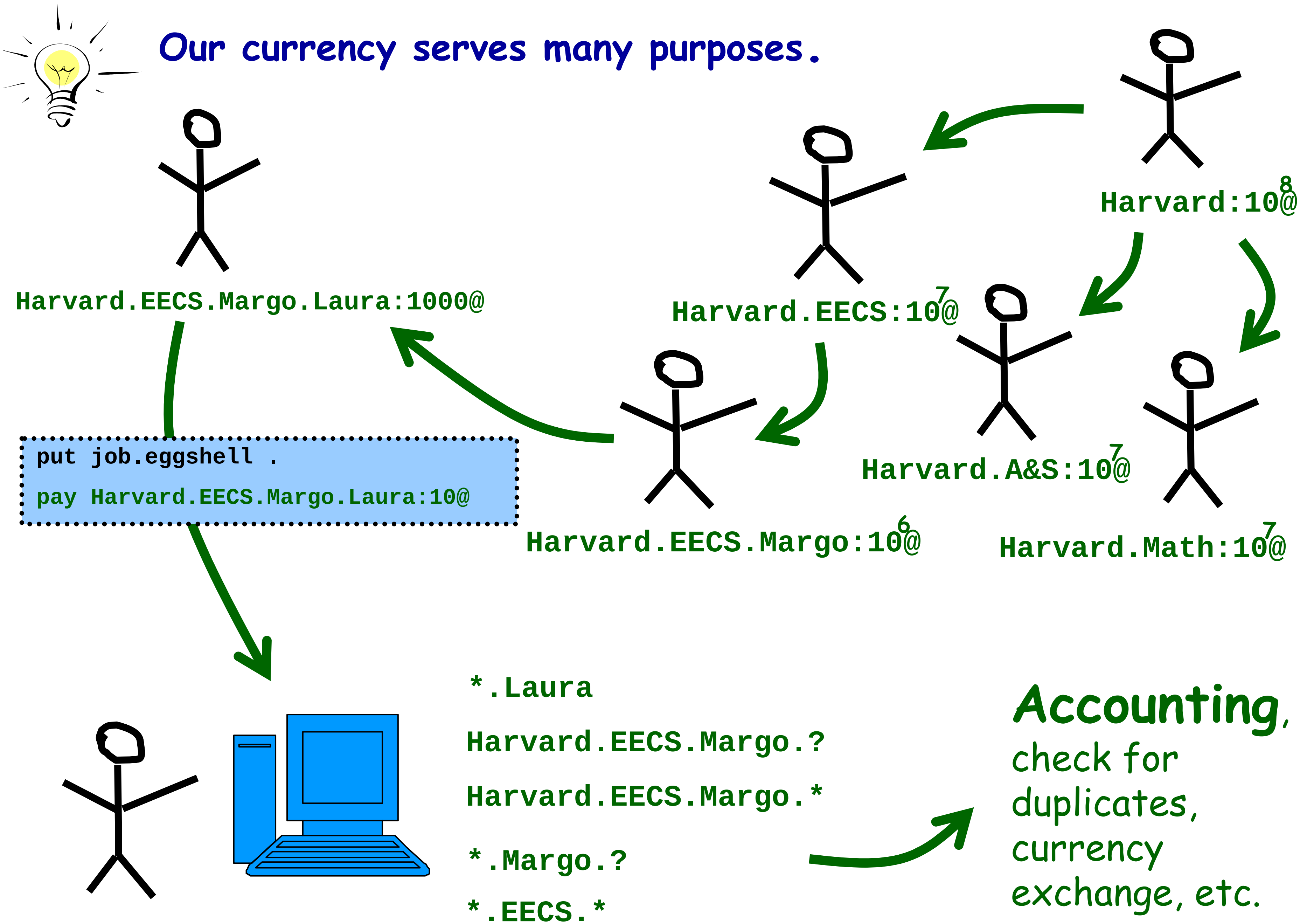


Motivating Scenario II: BU & Harvard

- Harvard and BU decide to share grid resources.
- Banks retain local currency. **Exchange rate.**
- HU caches annotated:
  - accept BU.* and HU.* currency
  - hard constraints on **x%** of BU jobs that can run
- BU caches similarly.
- Users at HU can use BU caches transparently. Maintain HU eggs. Exchange via BU bank.

- BU protected if HU decides to print a million HU eggs.

Our currency serves many purposes.

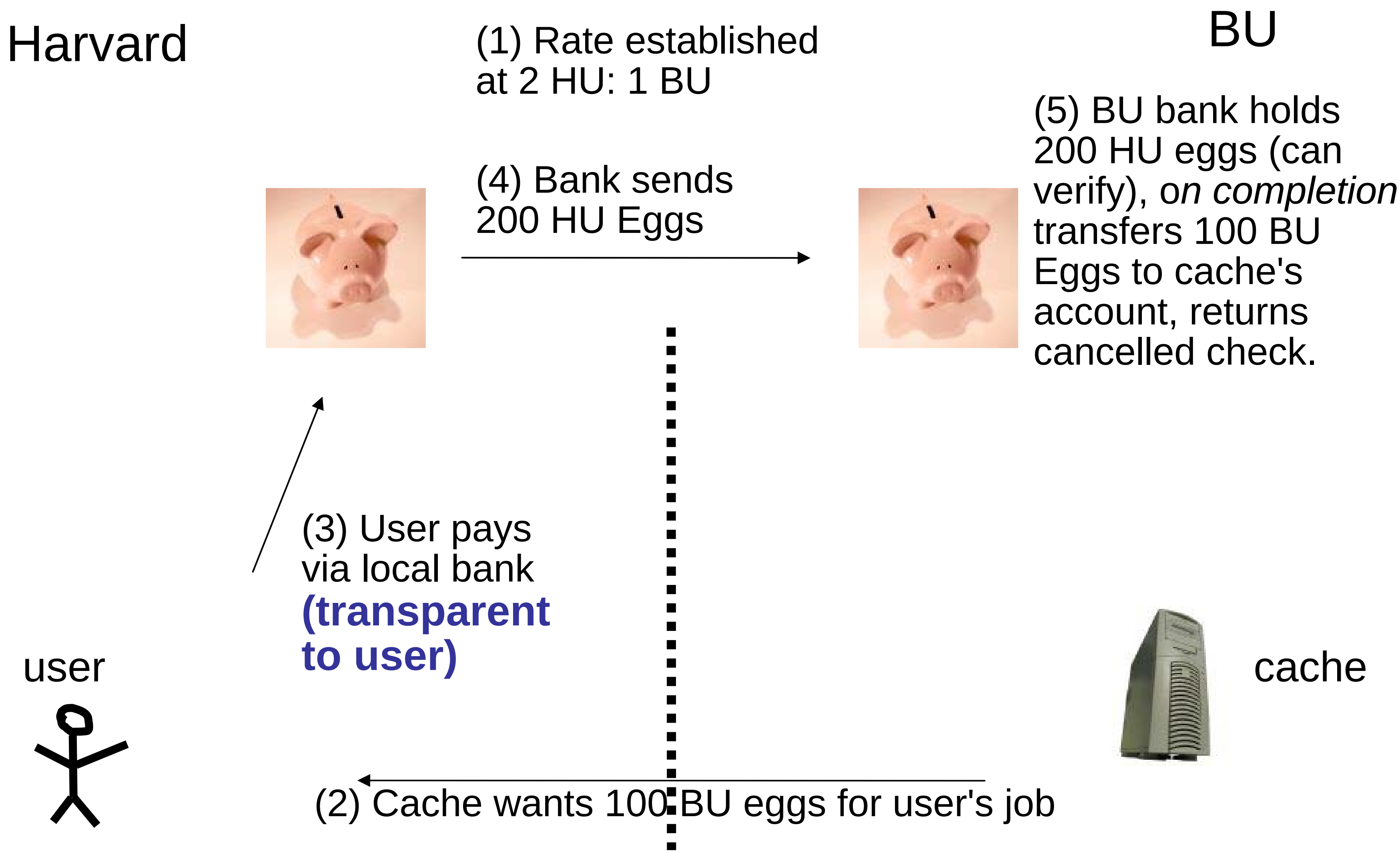


Anatomy of a piece of an Egg

Harvard.CS.Alice[Apr-02-06]:10.0@

- **Harvard**: Generating by the Harvard identity
- **CS**: Tranferred from Harvard to CS and signed by CS
- **Alice**: Transferred from CS to Alice and signed by Alice
- **[Apr-02-06]**: Expiration date
- **10.0**: Denomination

An Example of a Transaction



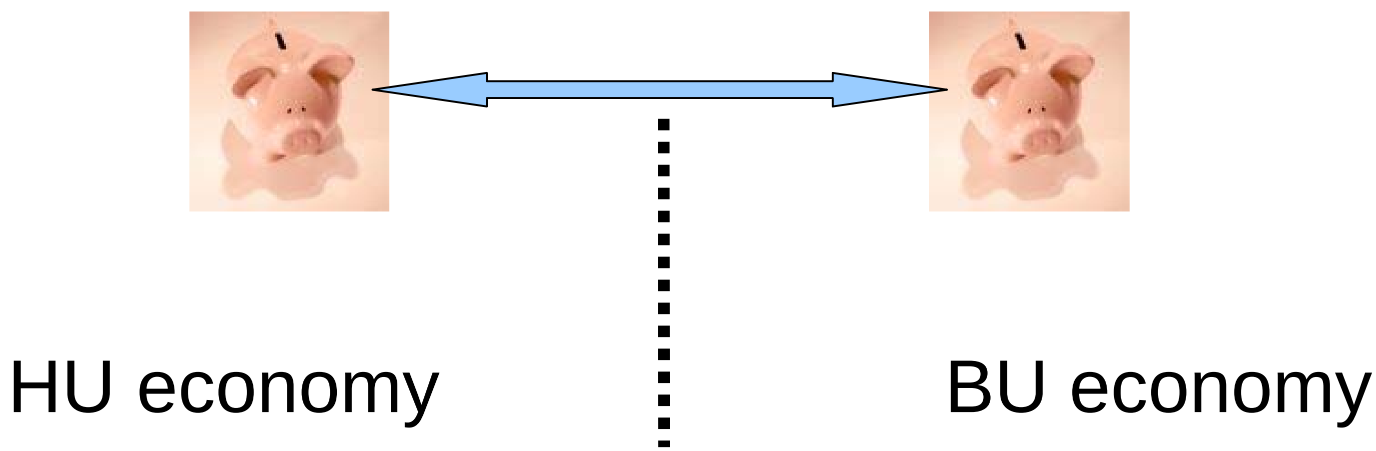
Exchange rates: Desirable Properties

- Facilitate cross-grid exchange
- Prevent abuse by autonomous economies
- Preserve value of currency used to pay for services
- Extensible: we provide first versions

Idea One: Based on 'Spending Power'

$$r_{bh} = \frac{c_h / q_h}{c_b / q_b}$$

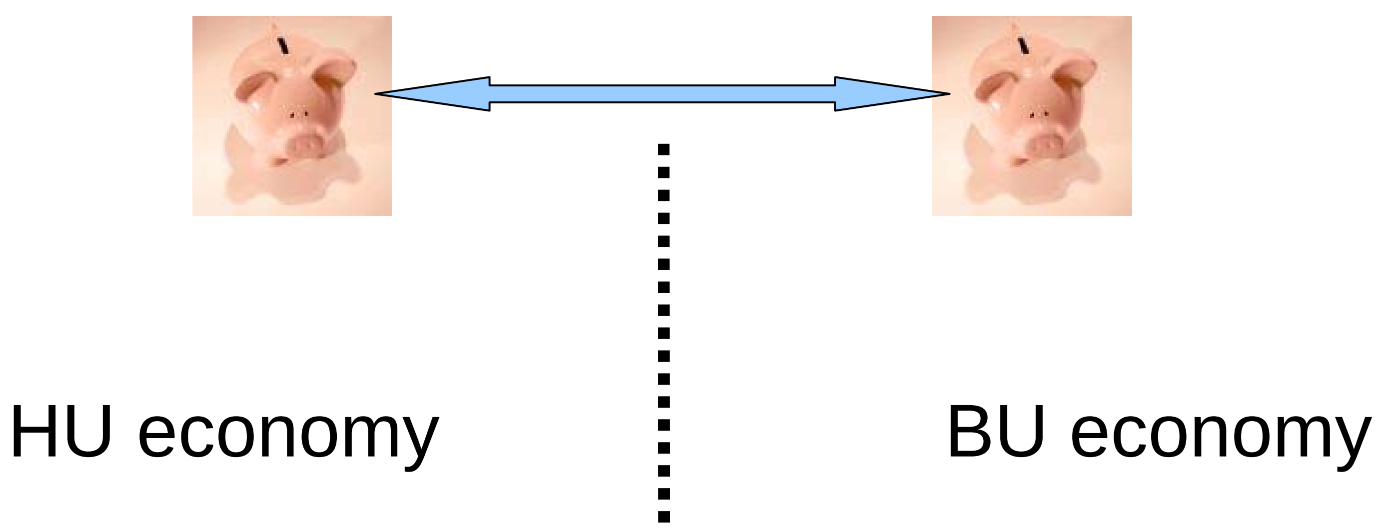
c_i : # currency
q_i: # resources
$x_b \cdot r_{bh} = x_h$



→ need to instrument economies
→ need to measure q_i across different resources [use relative prices w/in an economy, combined with some numeraire good. (or gold standard economy). real world GDP doesn't help.]

Idea Two: Trading Agents

Simple trading agents
Take positions in currency
Set prices to equilibrate supply and demand
Monitor positions. Swap out.
Willingness to take a position should depend on stability of economy
Lots of ideas hear from finance!



PPP: Test of Well-functioning system

Purchasing Power Parity

- In the long run, prices should move toward rates that equalize the prices of an identical basket of goods across countries

- **Requires**: basket is traded across borders, no tariffs, constant profit margins, same productivity,…

  ?? ok for grids

The hamburger standard

	Big Mac prices in local currency	Big Mac prices in dollars	Implied PPP* of the dollar	Actual dollar exchange rate April 22nd	Under (-)/over (+) valuation against the dollar, %
United States†	$2.71	2.71			
Argentina	Peso 4.10	1.43	1.51	2.88	-47
Australia	A$3.00	1.86	1.11	1.61	-31
Brazil	*Real* 4.55	1.48	1.68	3.07	-45
Britain	£1.99	3.14	1.36‡	1.58‡	+16
Canada	C$3.20	2.21	1.18	1.45	-18
Chile	Peso 1,400	1.95	517	716	-28
China	Yuan 9.90	1.20	3.65	8.28	-56
Czech Rep	Koruna 56.57	1.96	20.9	28.9	-28
Denmark	DKr27.75	4.10	10.2	6.78	+51
Egypt	Pound 8.00	1.35	2.95	5.92	-50
Euro area	€2.71	2.97	1.00§	1.10§	+10
Hong Kong	HK$11.50	1.47	4.24	7.80	-46
Hungary	Forint 490	2.18	181	224	-19
Indonesia	Rupiah 16,100	1.84	5,941	8,740	-32
Japan	¥262	2.19	96.7	120	-19
Malaysia	M$5.04	1.33	1.86	3.80	-51

The Economist, **April 2003**

Conclusions

- Egg provides an extensible and economics-inspired grid computing platform
- Close-collaboration between CS, physics, economics
- Spawning many subprojects
  - statistical machine learning for resource prediction
  - open MD for sequential environments
  - opportunity-cost based schedulers
  - algorithms to compute exchange rates
  - languages for environment computing
- Extensible + Open:: Define and implement platform, first versions of various caches. Continual innovation (by anyone) will be supported and encouraged.

www.eecs.harvard.edu/econcs