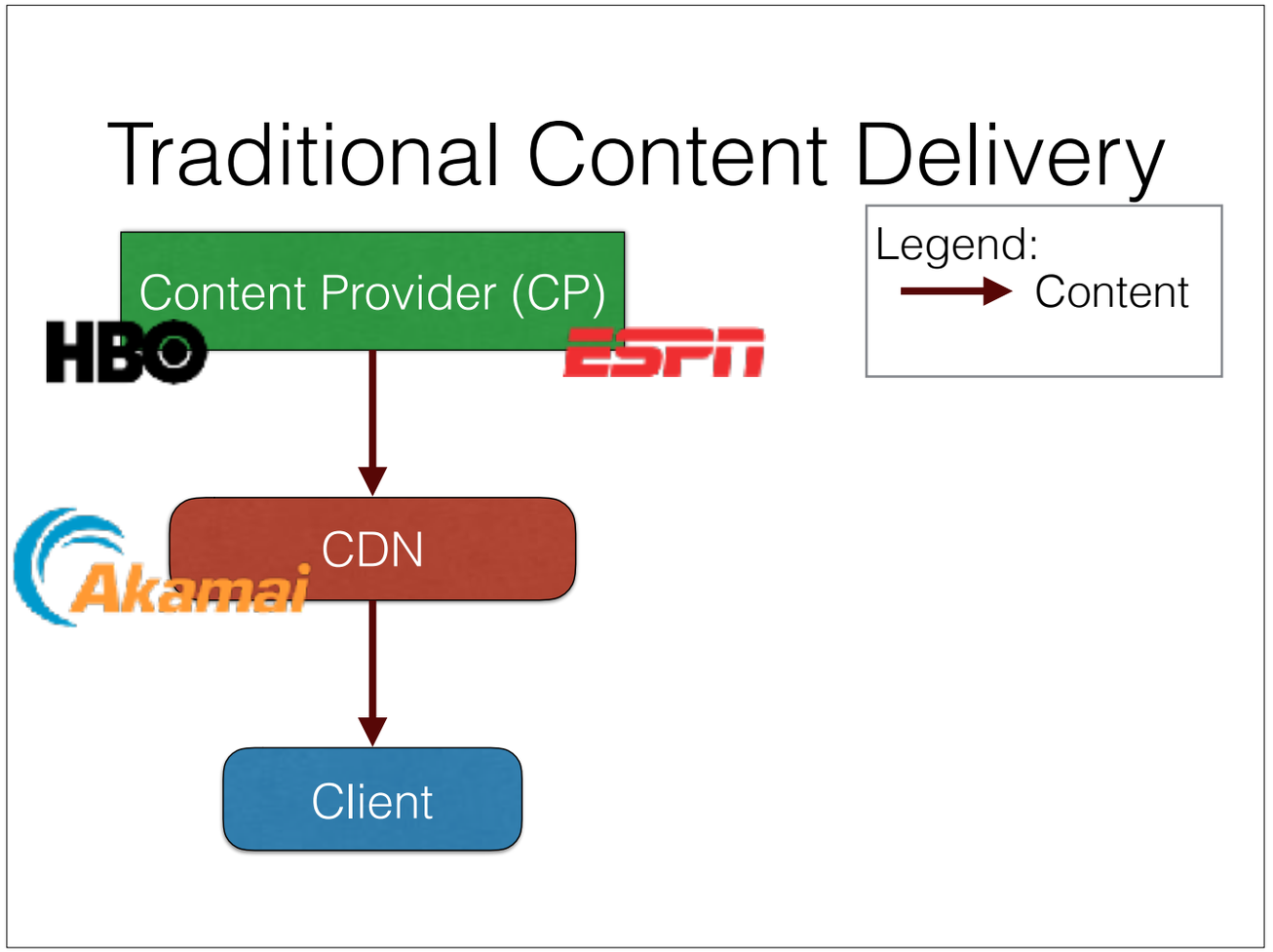
Redesigning CDN-Broker Interactions for Improved Content Delivery

Matthew K. Mukerjee, I. Nadi Bozkurt, Devdeep Ray,
Bruce Maggs, Srinivasan Seshan, Hui Zhang

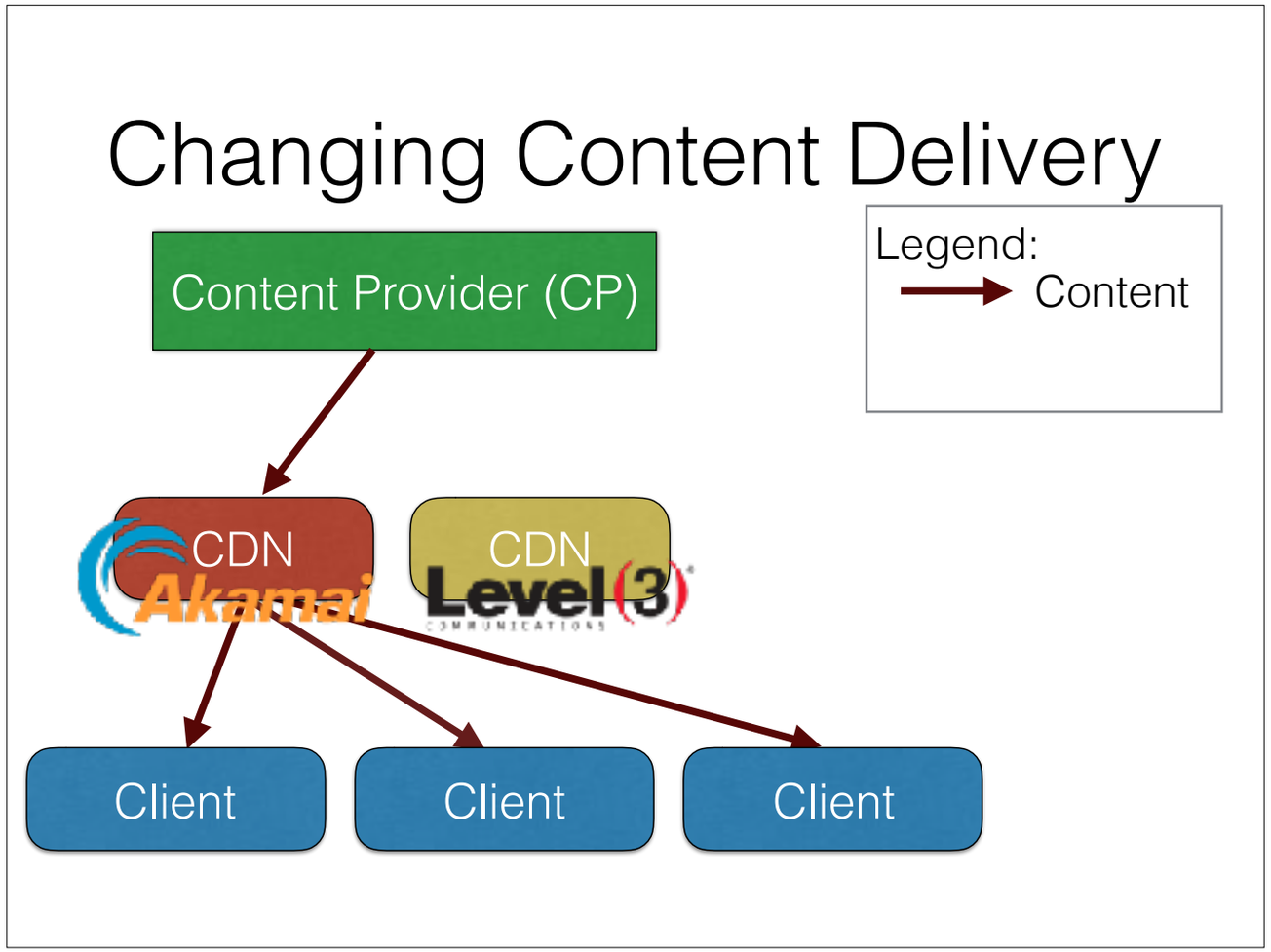
CoNEXT ‘17

Carnegie Mellon University Duke University Akamai CONVIVA

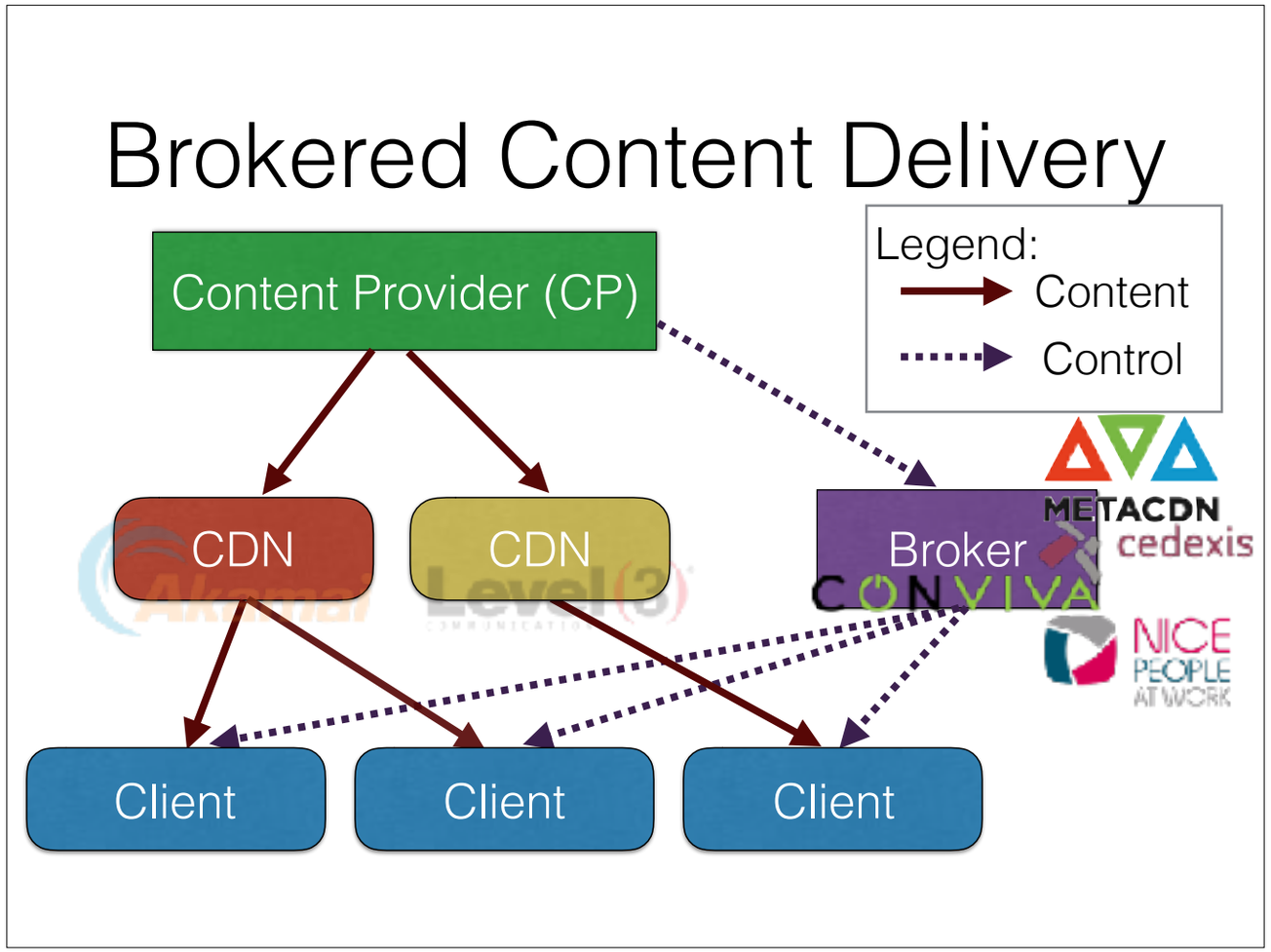
Hi, my name is Matt Mukerjee and I’ll be presenting our work on “Redesigning CDN-Broker Interactions for Improved Content Delivery.”



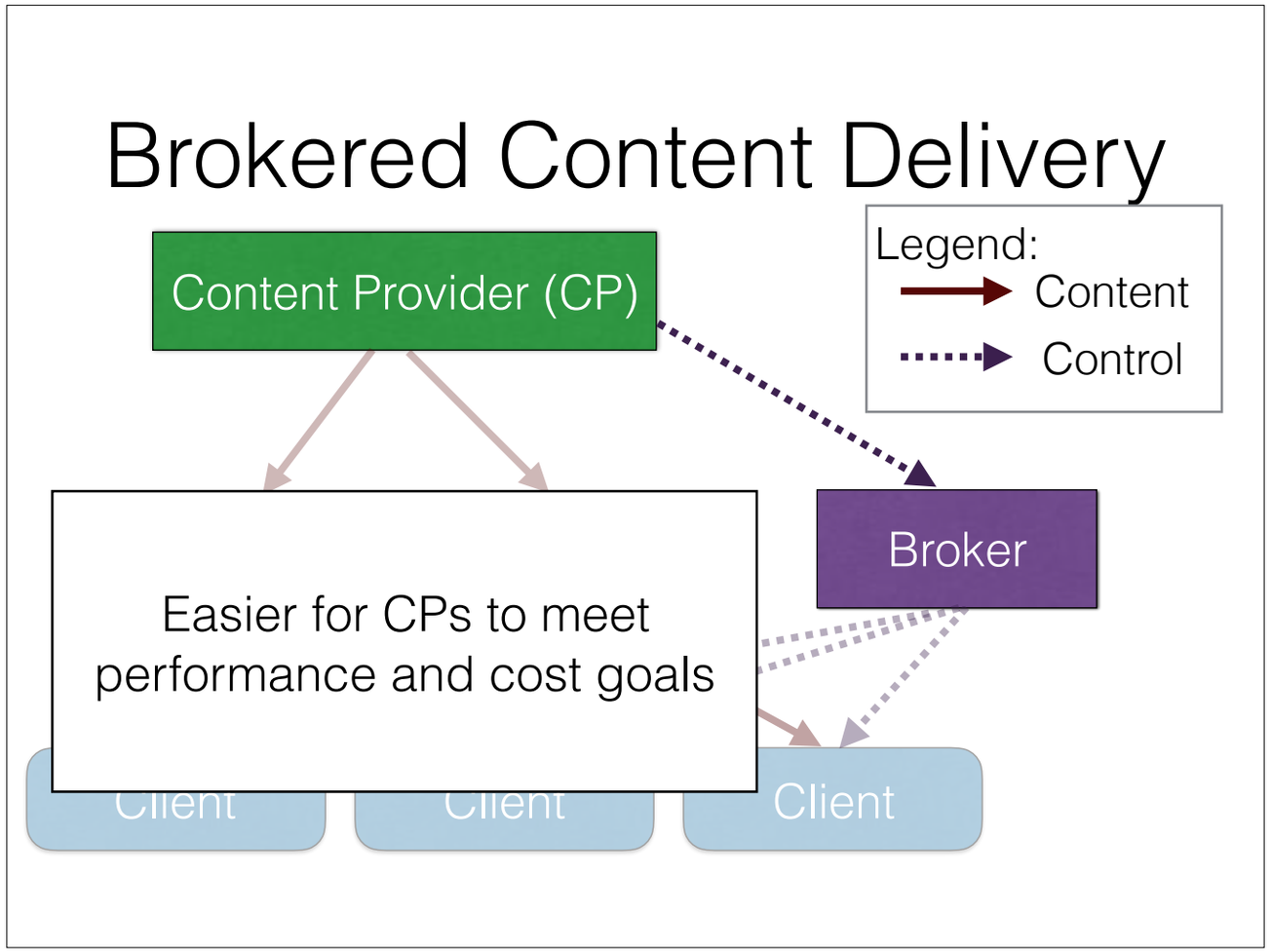
Traditional content delivery involves content providers (like ** HBO and ** ESPN), sending their content to CDNs (like ** Akamai), which ultimately deliver the data to clients. The picture is complicated by…



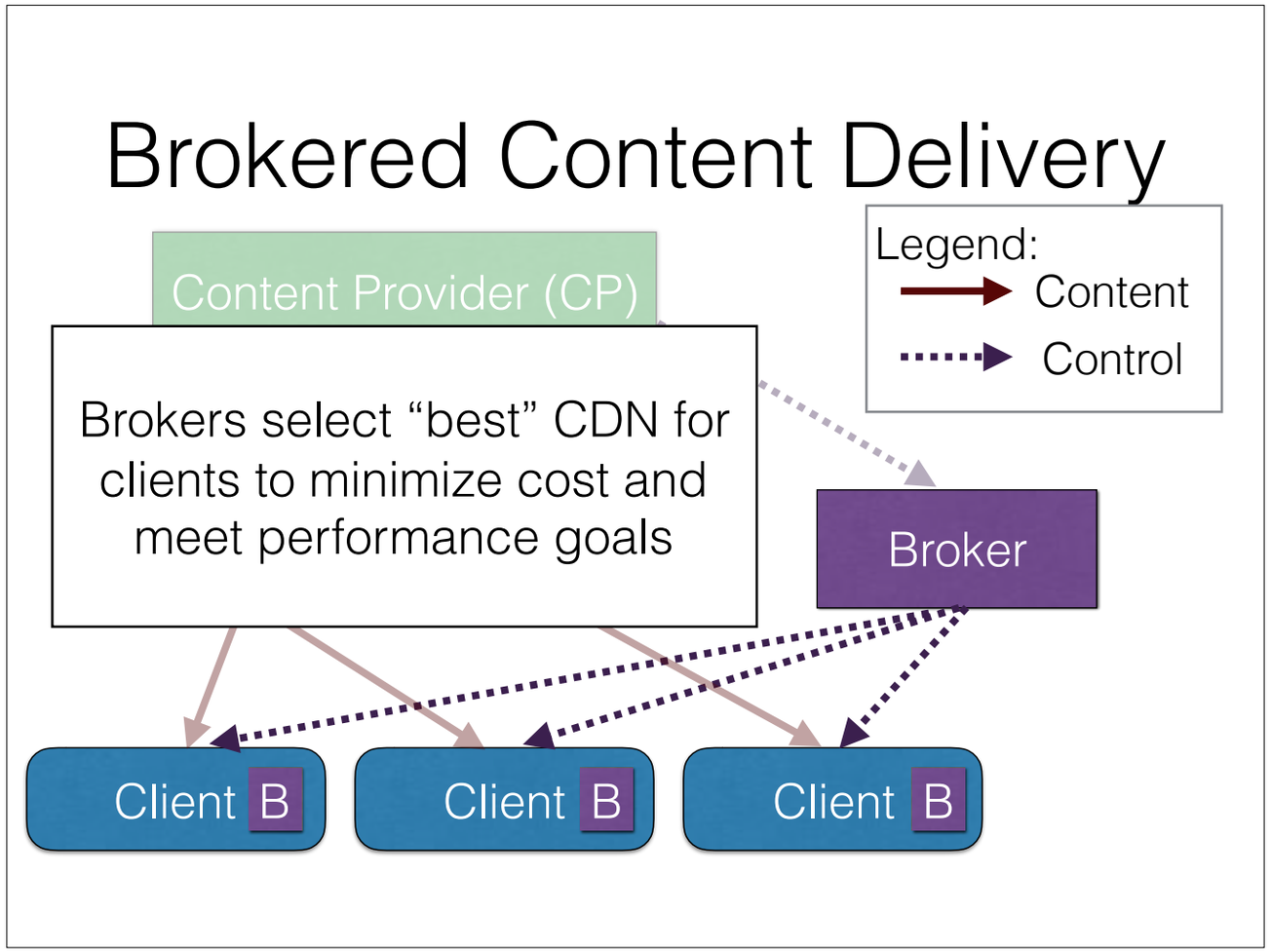
… many clients as well as ** other CDNs **. ** In order to make better use of the opportunities offered by stitching together multiple CDNs, an additional entity is involved in content delivery today, …



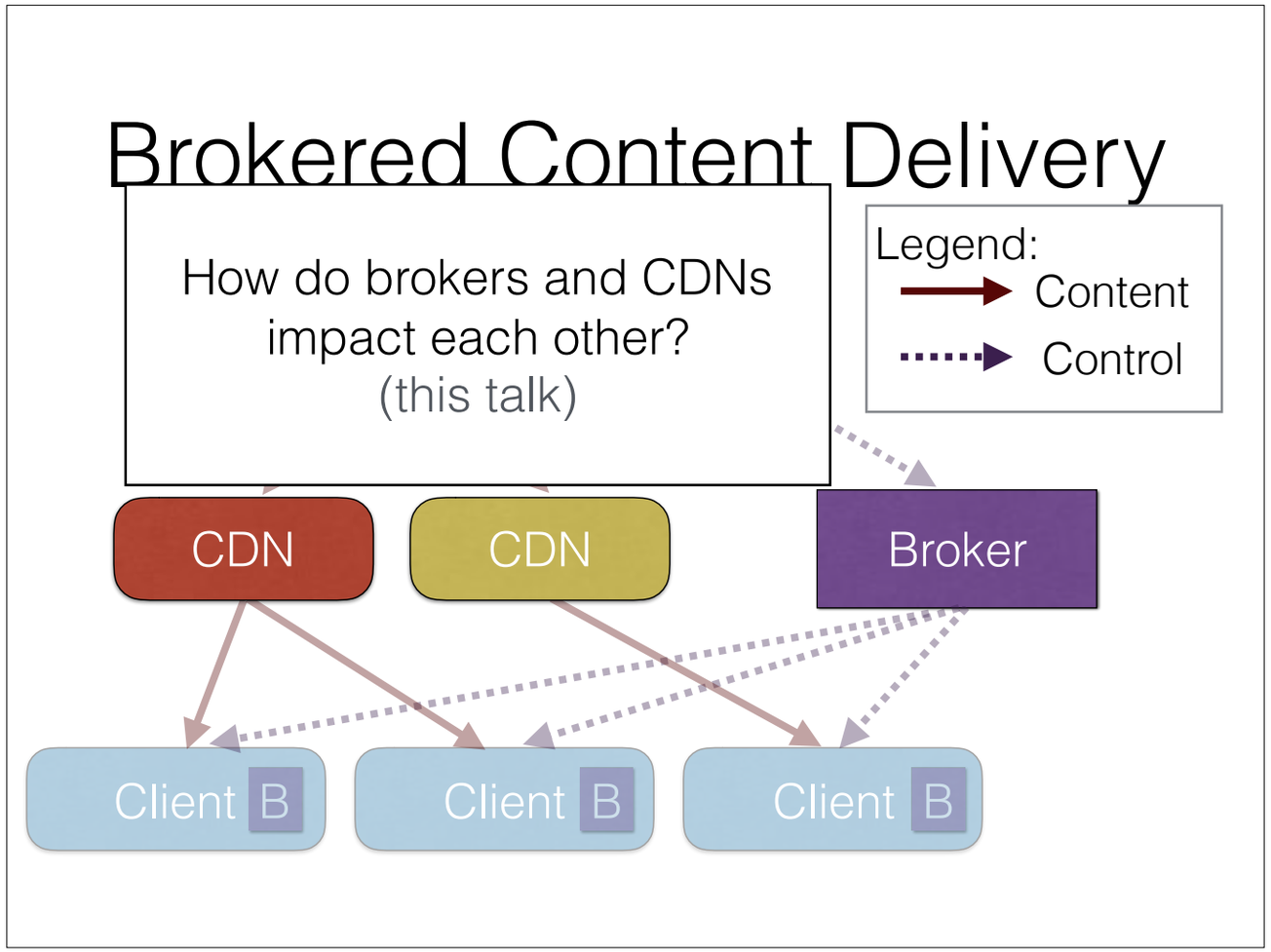
… called a broker (** e.g., Conviva, Cedexis, MetaCDN, etc.). ** Brokers are purely a control plane entity that stitch together CDNs, …



… making it easier for content providers to meet performance and cost goals.



They do so by selecting the appropriate CDN for clients. Brokers run software on the clients (e.g., a video player on ESPN’s website) that contact the broker periodically to select the “best” CDN for the client based on things like device type, geographic location, and ISP. The “best” CDN may change over time.



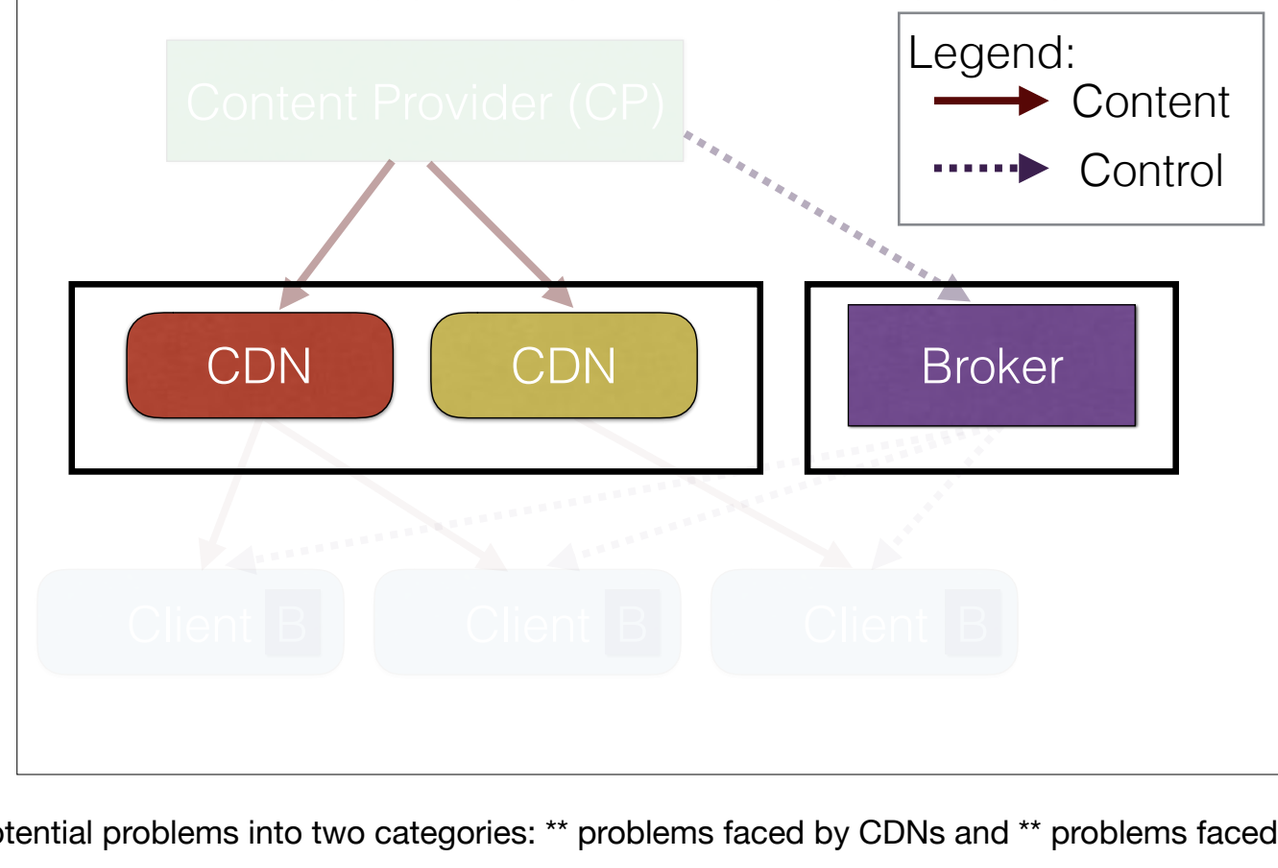
What we don’t understand well is how the decisions made by the broker affect the decisions made by the CDNs and vice-versa. To exacerbate this— currently brokers and CDNs don’t have an interface; they don’t explicitly communicate with each other to make decisions, potentially leading to problems.

Contributions

- Identify challenges that brokers and CDNs create for each other by analyzing data from both
- Examine the design space of CDN-broker interfaces
- Evaluate the efficacy of different designs

In this work, ** we identify these problems by analyzing data from both, ** examine the design space of CDN-broker interfaces, and ** evaluate the efficacy of the different designs in the design space.

Potential Problems



First— potential problems: we group potential problems into two categories: ** problems faced by CDNs and ** problems faced by brokers. (Let’s dig into these)

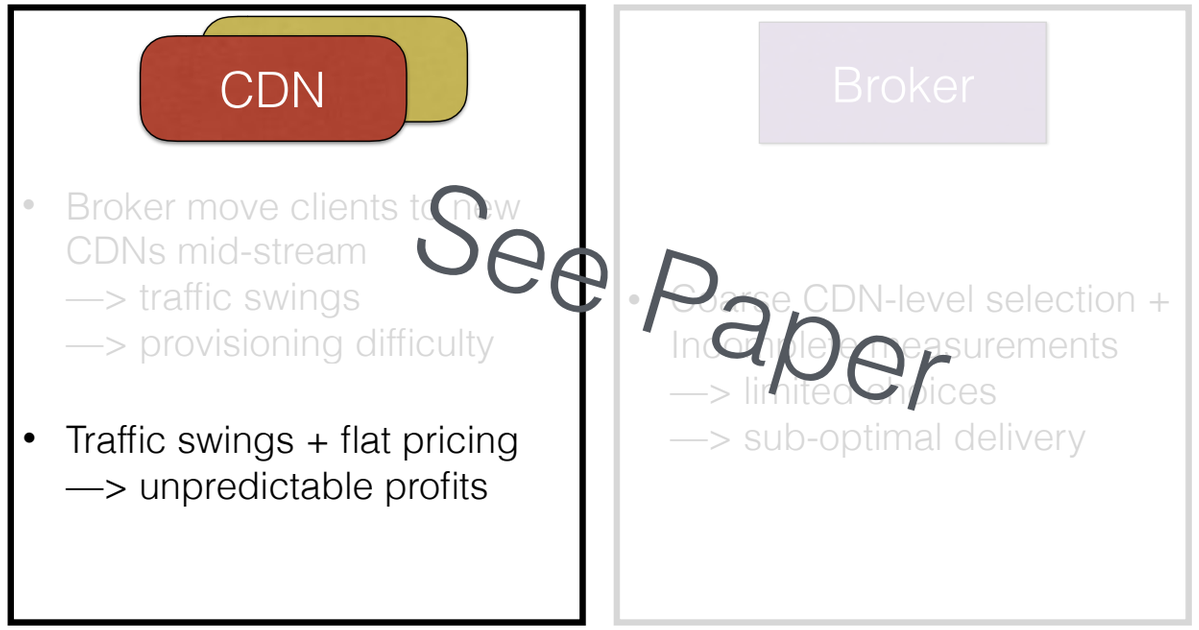
Potential Problems

- Broker move clients to new CDNs mid-stream
  —> traffic swings
  —> provisioning difficulty
- Traffic swings + flat pricing
  —> unpredictable profits

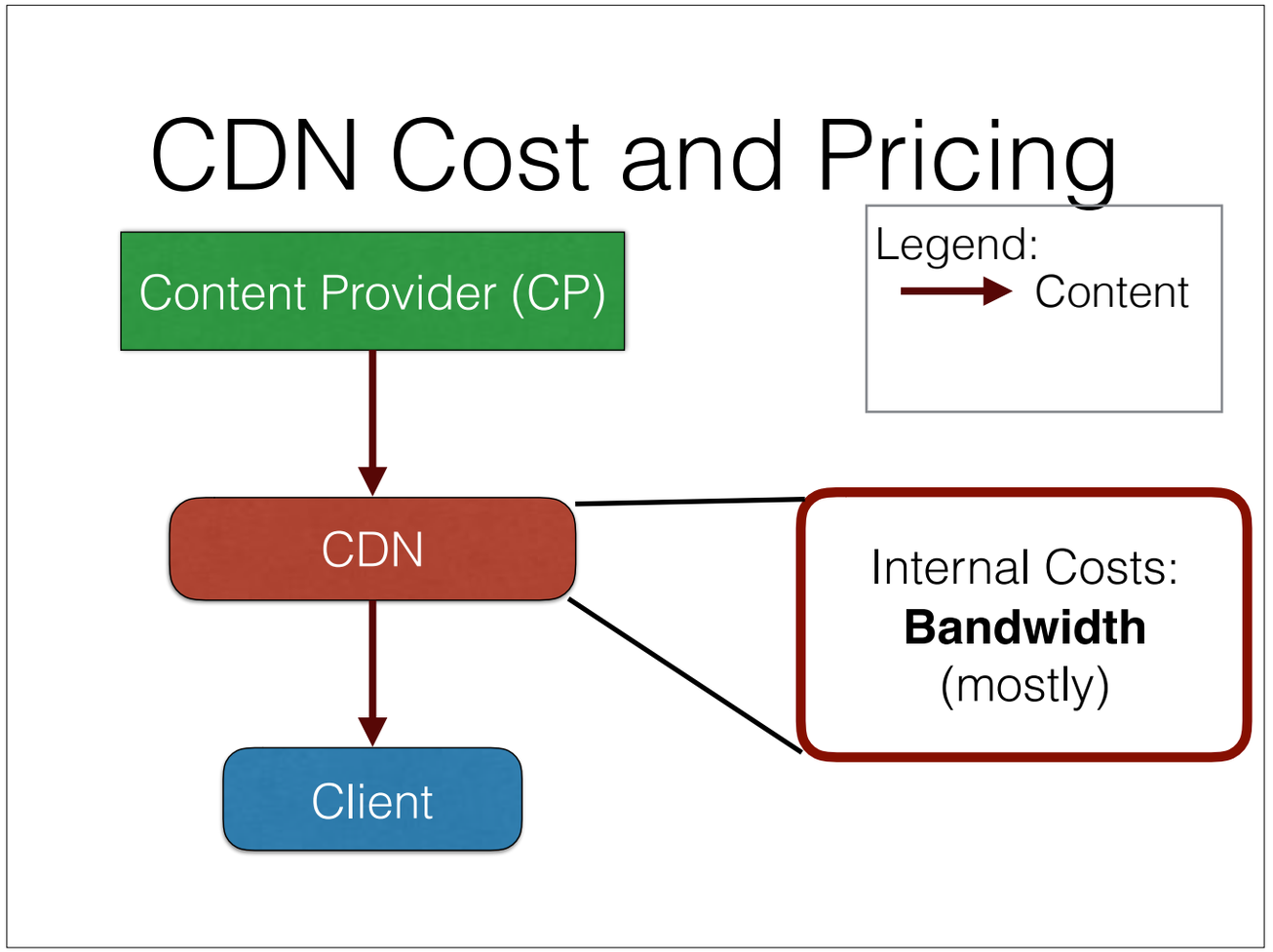
- Coarse CDN-level selection + Incomplete measurements
  —> limited choices
  —> sub-optimal delivery

On the CDN-side, brokers move many clients to different CDNs mid-stream leading to rather large traffic swings. This could complicate CDN provisioning. ** With these traffic swings, CDNs flat pricing model makes profits unpredictable. Brokers face difficulty ** due to coarse CDN-level selection and incomplete measurements giving them limited choices to meet content provider goals. In this talk, we're going to only…

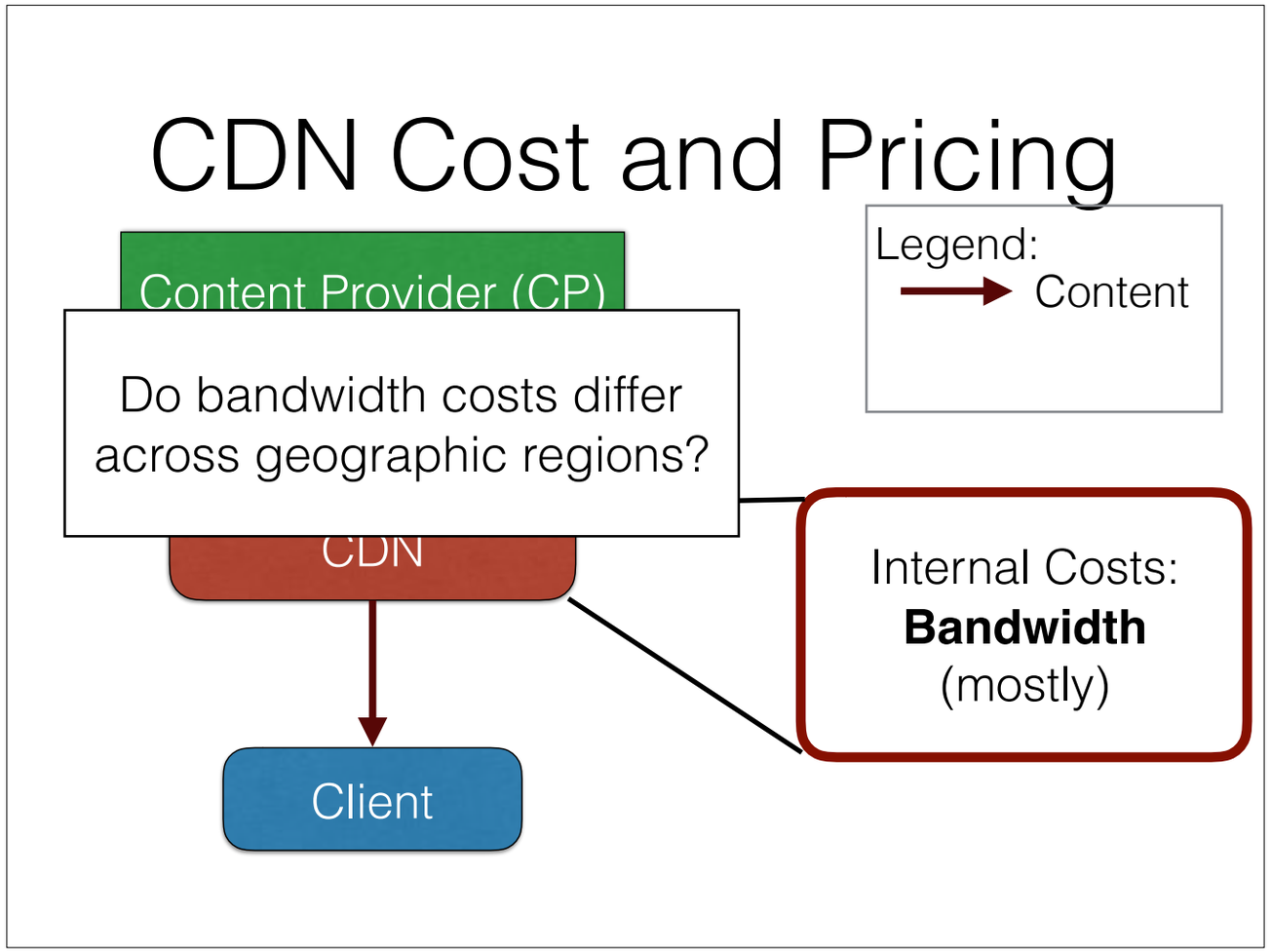
Potential Problems



… focus on one of the problems CDNs face. For insight into other problems, read through our paper. Okay, let's look at how traffic swings and flat pricing make CDN profits unpredictable.



To understand CDN profits, we need to understand their internal cost and revenue. ** We were told by a large CDN that their internal cost comes predominantly from paying ISPs for bandwidth.



A natural question is if bandwidth costs differ across geographic regions.

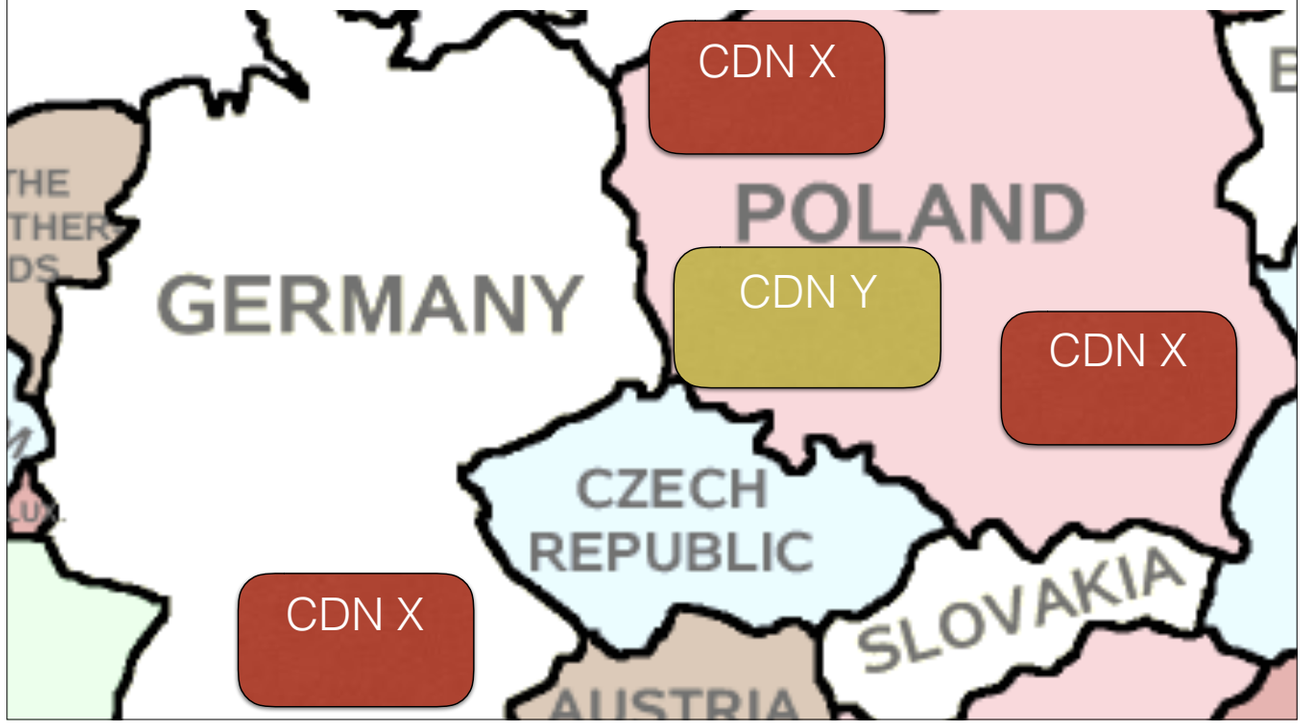
CDN Cost / Byte Delivered

30x

difference in cost per byte
between the most expensive
and least expensive countries

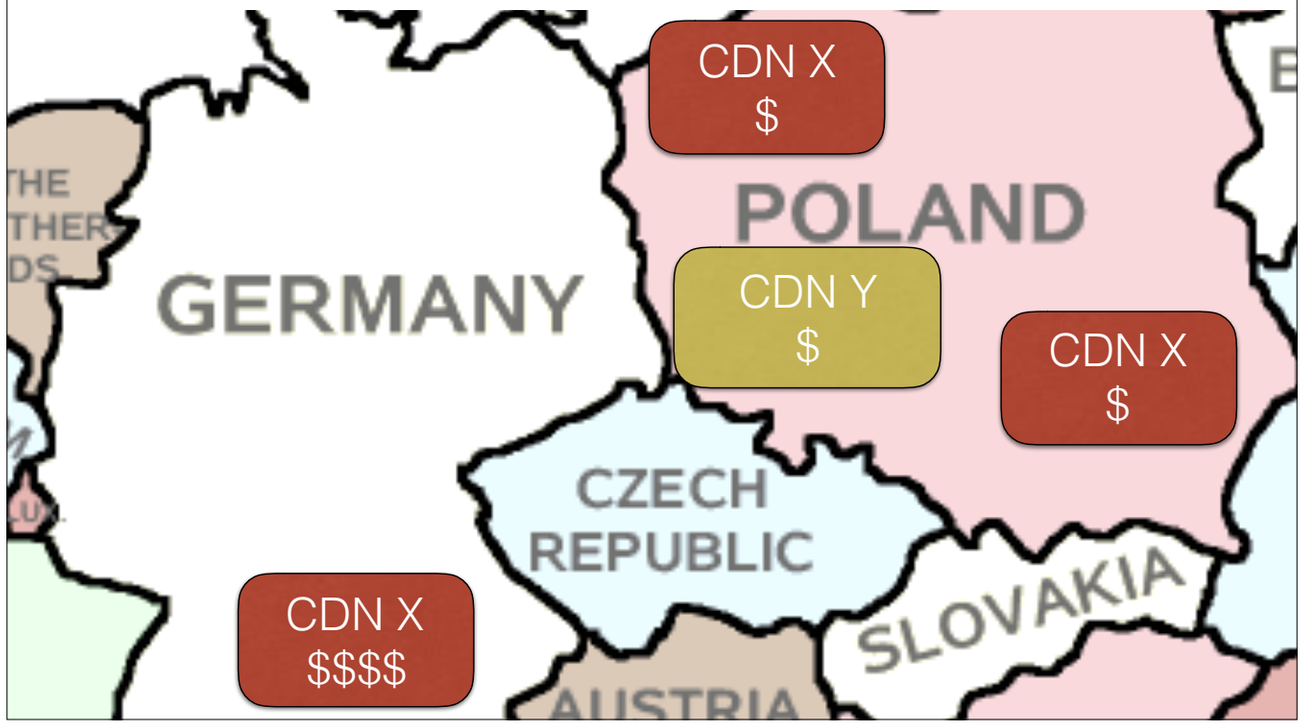
We got data on the cost per byte delivered from a major CDN for the top 20 countries with the most requests. There was a ** 30 times difference in cost between the most expensive and least expensive country. I want to explain how we’re going to represent these internal costs, so let’s go through another hypothetical example.

CDN Internal Cost

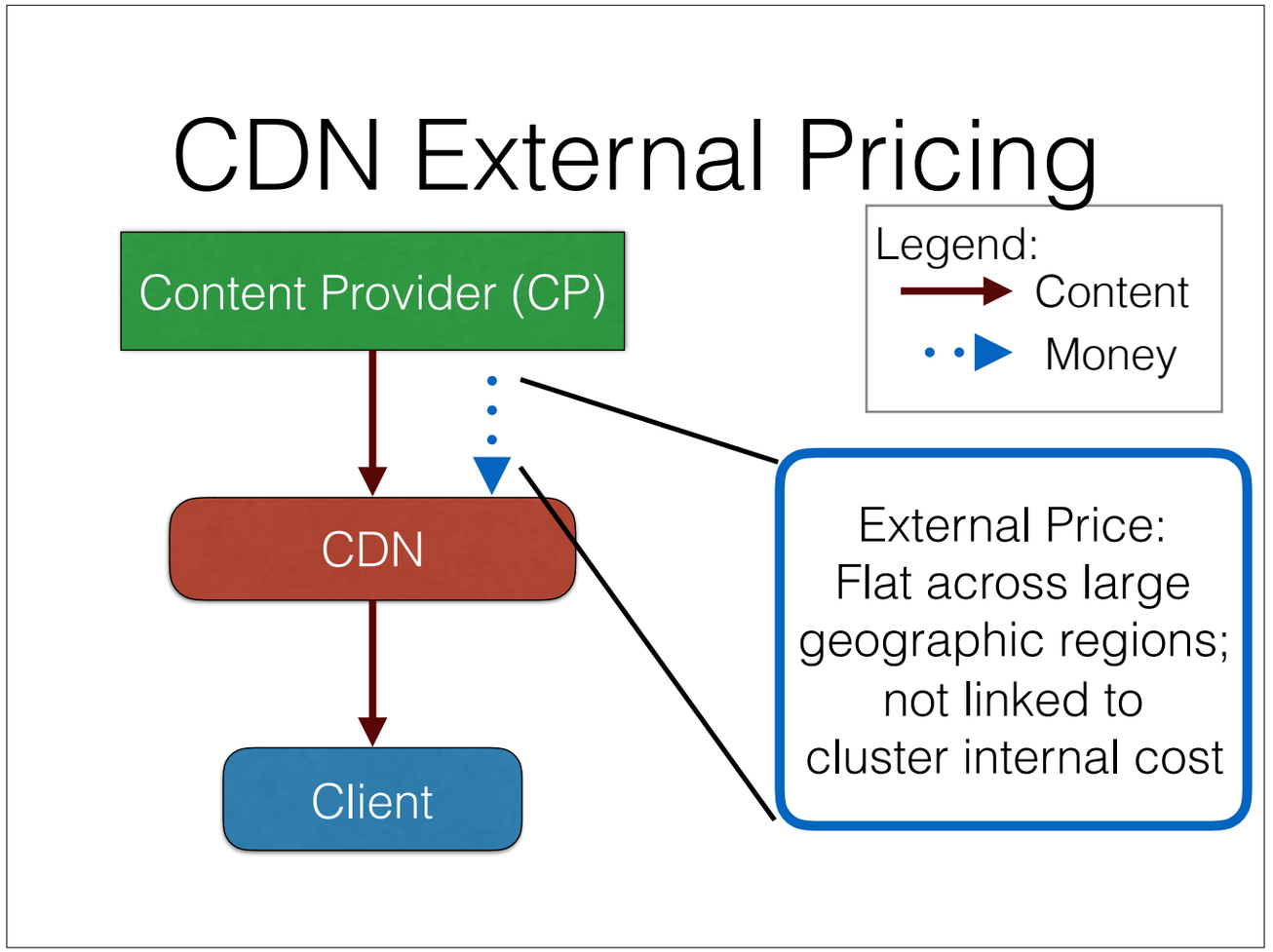


Here we see a zoomed in map of Europe. We have ** ** two CDNs, one with multiple clusters in different countries. We’re going to represent the delivery cost for individual CDN clusters as dollar signs listed on each cluster.

CDN Internal Cost

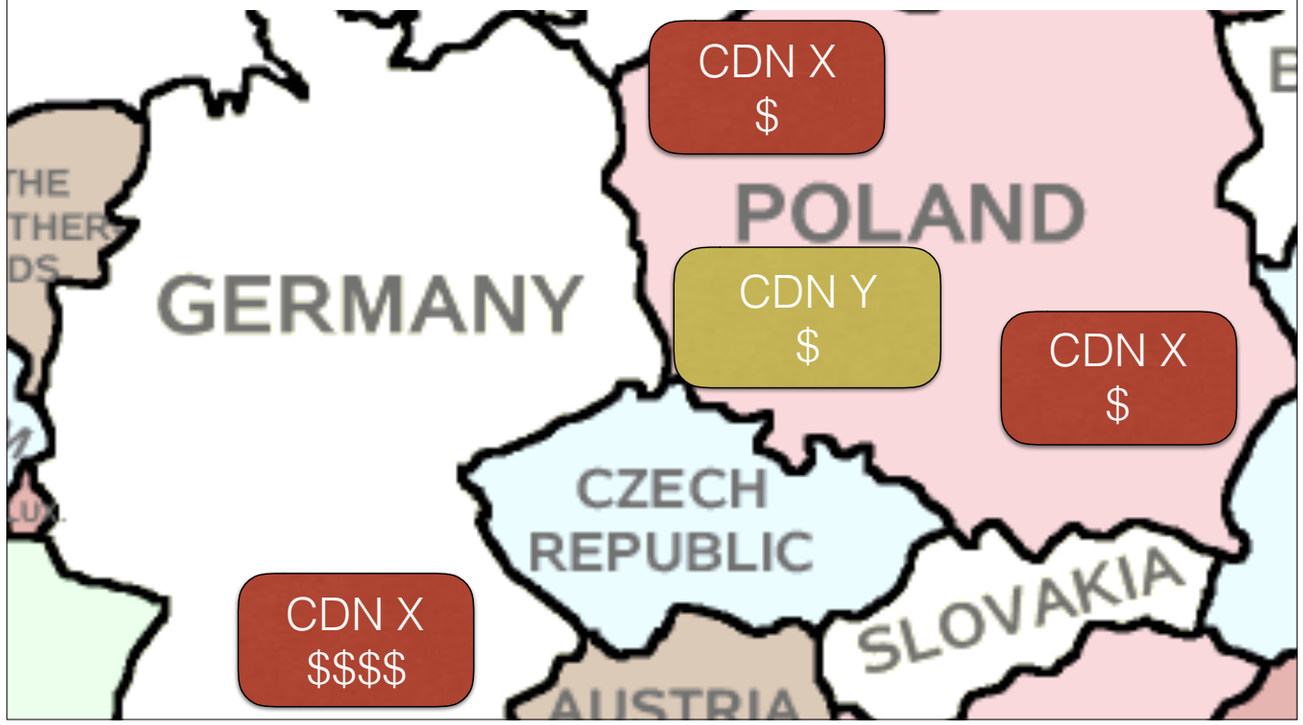


So here we see all the clusters in Poland are cheap to deliver from, but the cluster in Germany is very expensive. With that picture in mind, let's know learn how CDNs price their services.



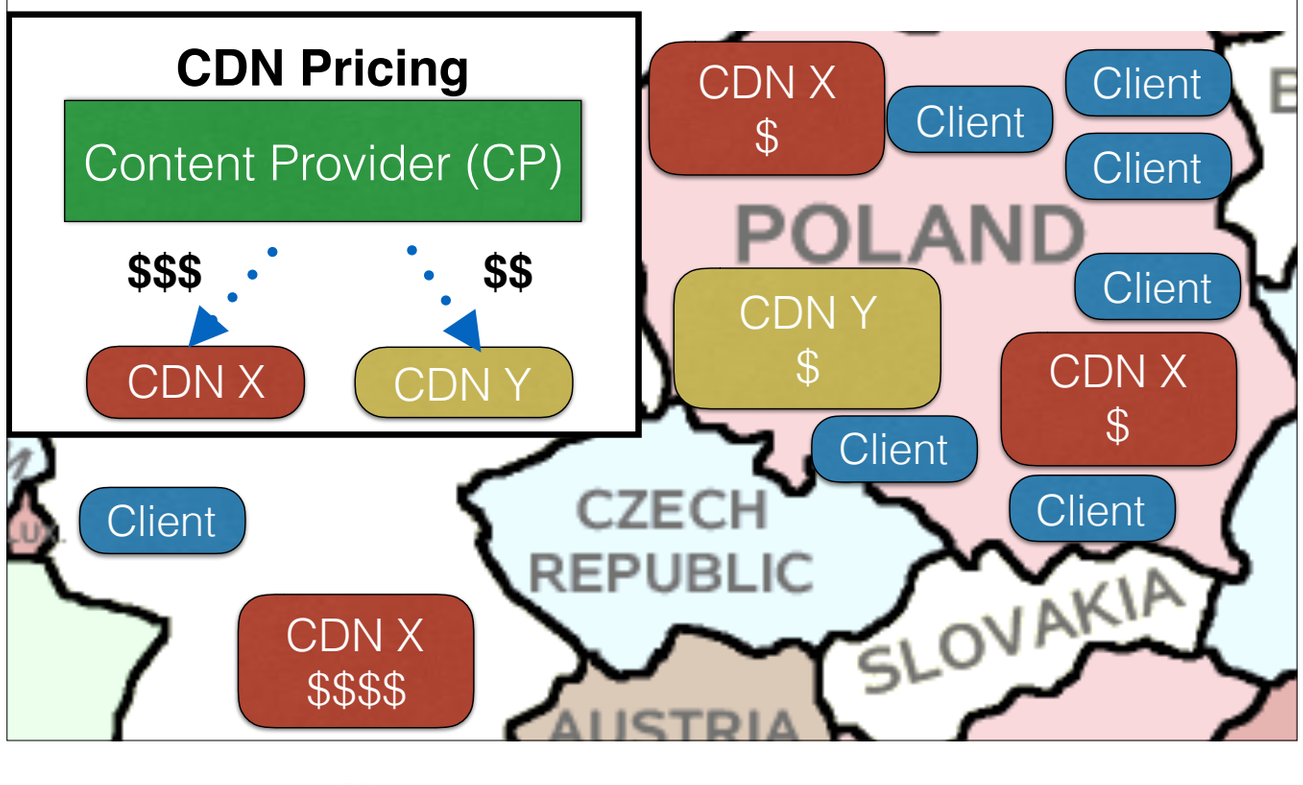
Content providers ** pay CDNs for their delivery services. CDNs negotiate their prices with content providers via long-term contracts, ** which generally have fixed prices across large geographic regions (e.g., continents). The key point is that these prices don’t relate to specific clusters’ delivery cost.

CDN External Price



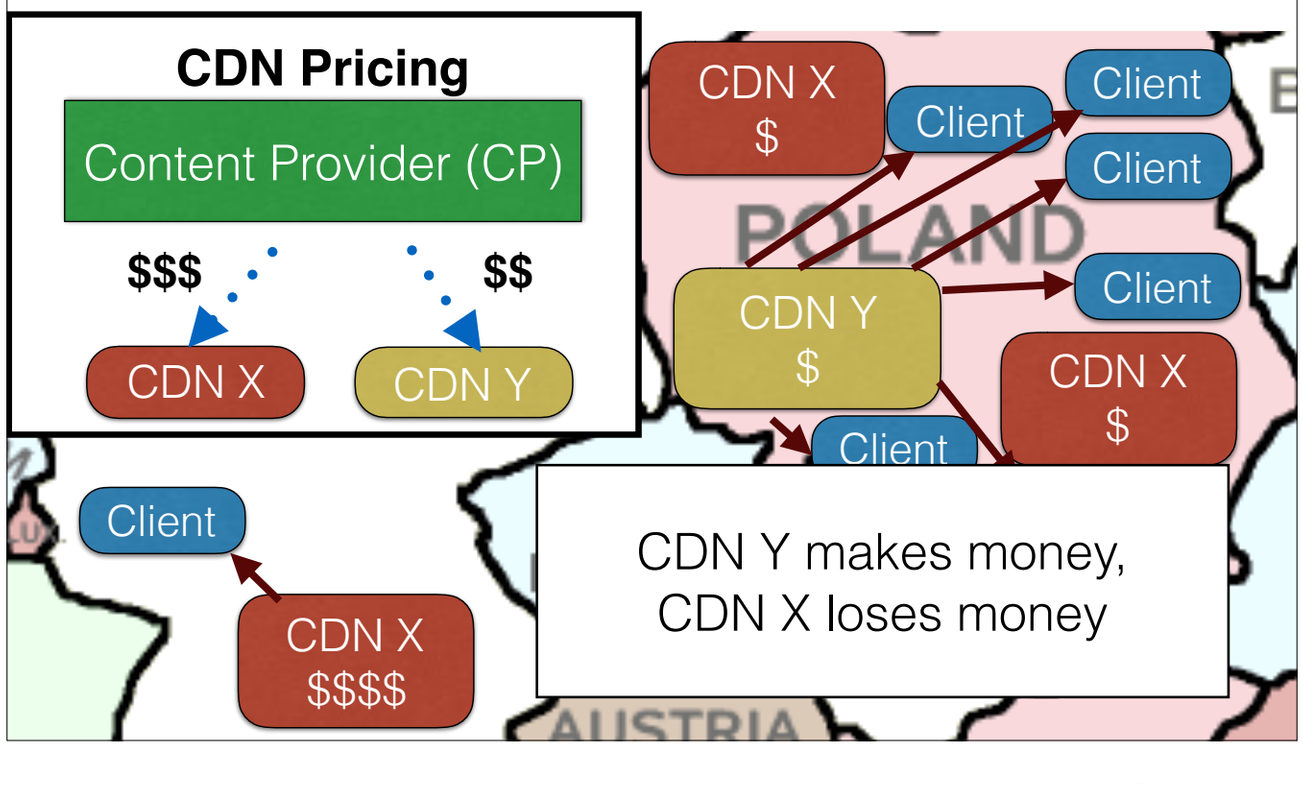
Going back to our hypothetical example; let’s say …

CDN External Price



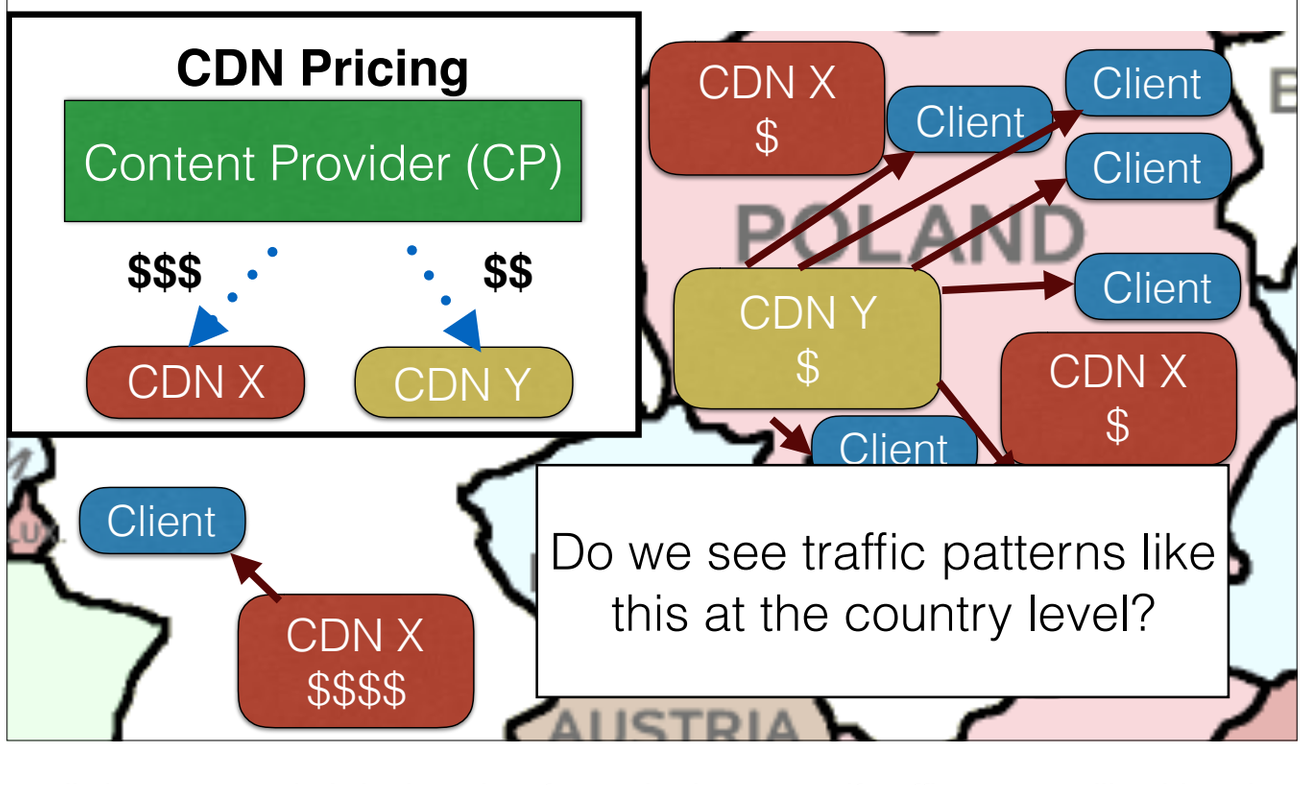
… the content provider negotiated the following contracts with CDN X and Y: it will pay X three dollar signs per byte delivered and Y two dollar signs per byte delivered. As I mentioned before, these are flat rates across whole continents. Now let’s bring in some clients **. The broker might allocate them something …

CDN External Price

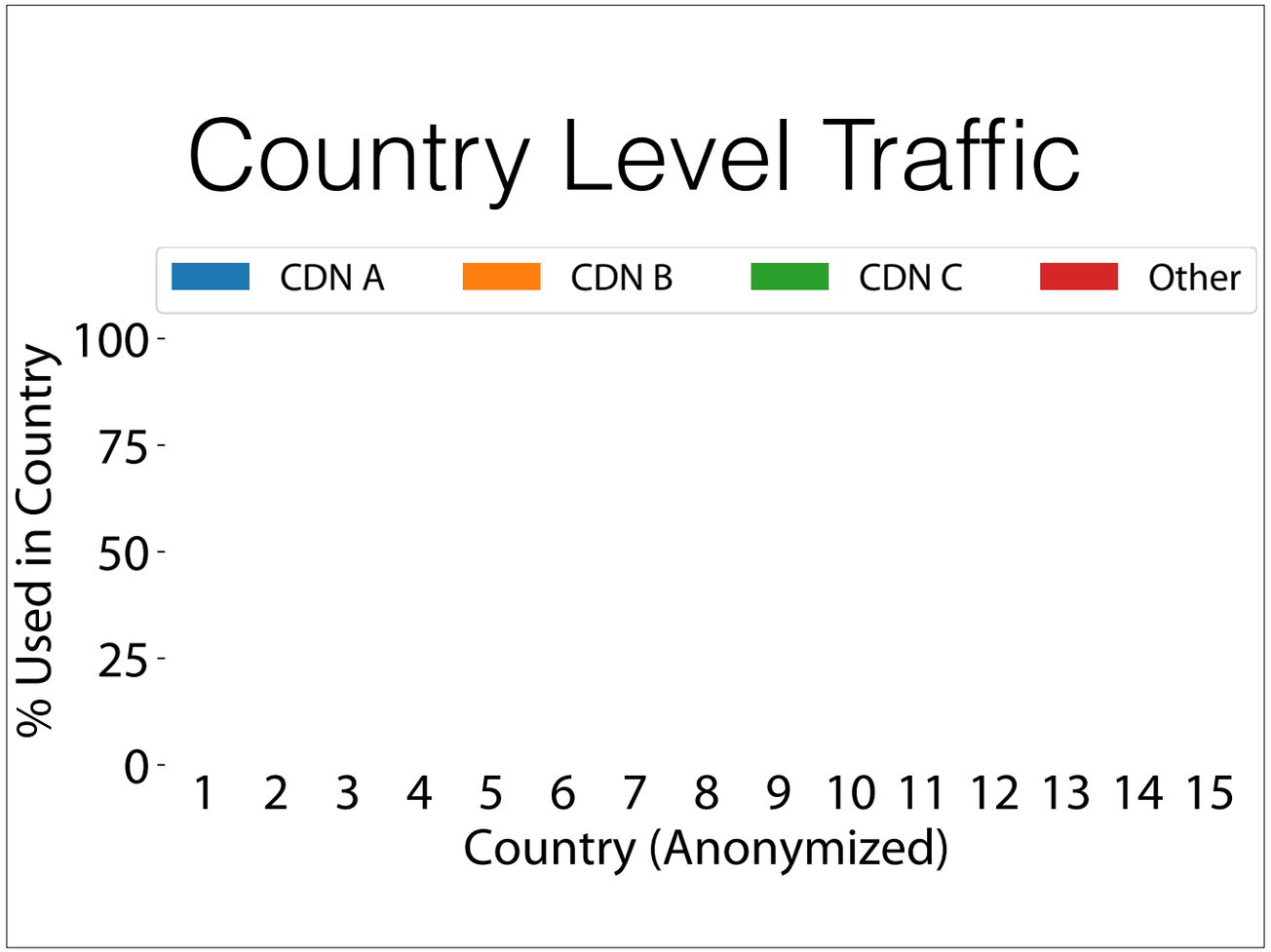


… like this. All clients in Poland go to CDN Y as it's cheaper and can provide adequate performance, and the client in Germany goes to CDN X as it is the only option that can provide adequate performance. Clearly, ** CDN Y makes money as its spends one dollar sign on delivery, yet charges the content provider two dollar signs. However, CDN X loses money as it is charging three dollar signs to the content provider, but only delivers data from it's expensive four dollar sign German cluster. If some of its cheaper Poland clusters were used it could make money, but they are avoided in favor of the cheaper CDN Y.

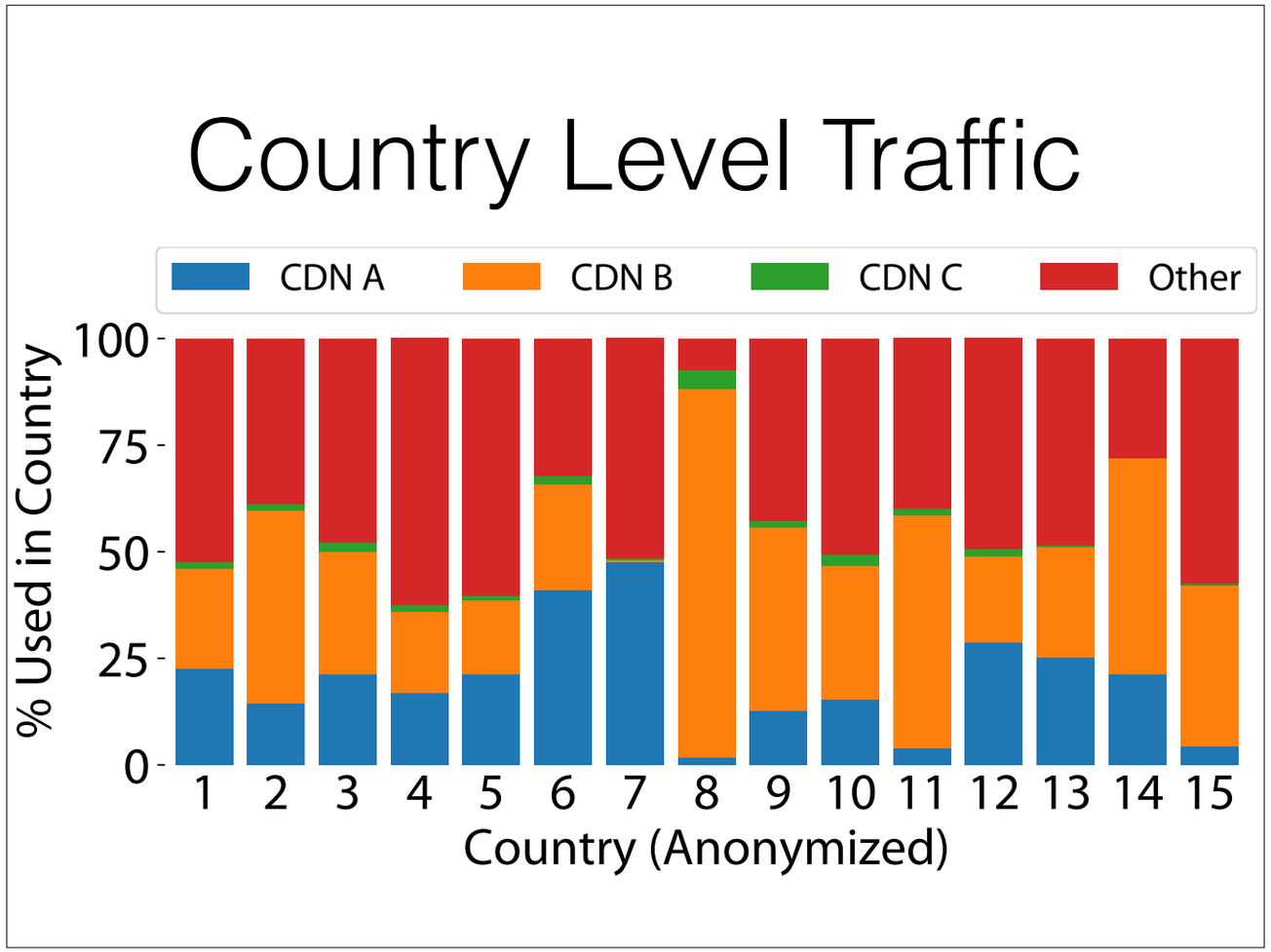
CDN External Price



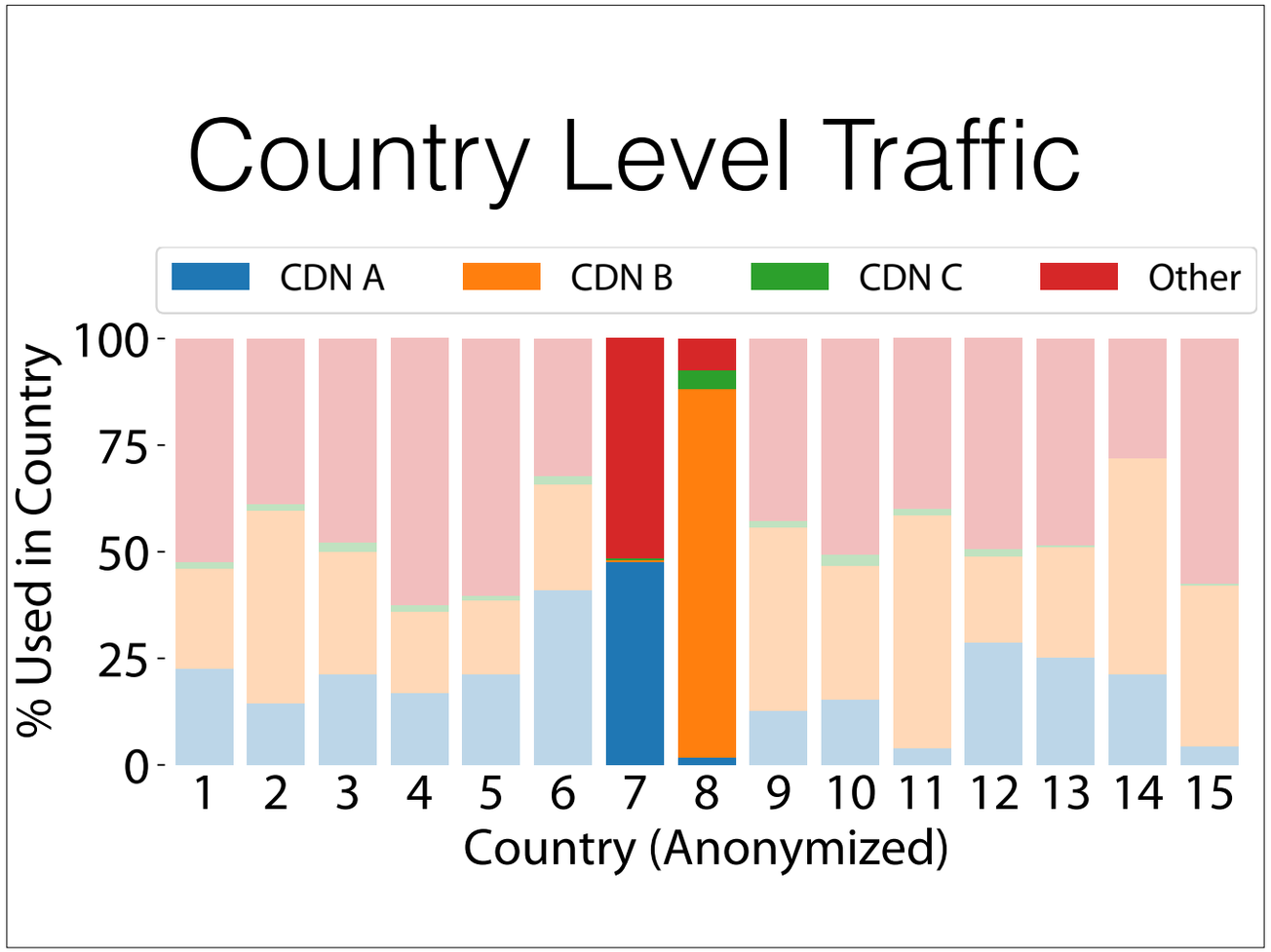
We want to know if something like this actually happens, so let’s look at data from a broker to see if traffic patterns like this exist at the country level.



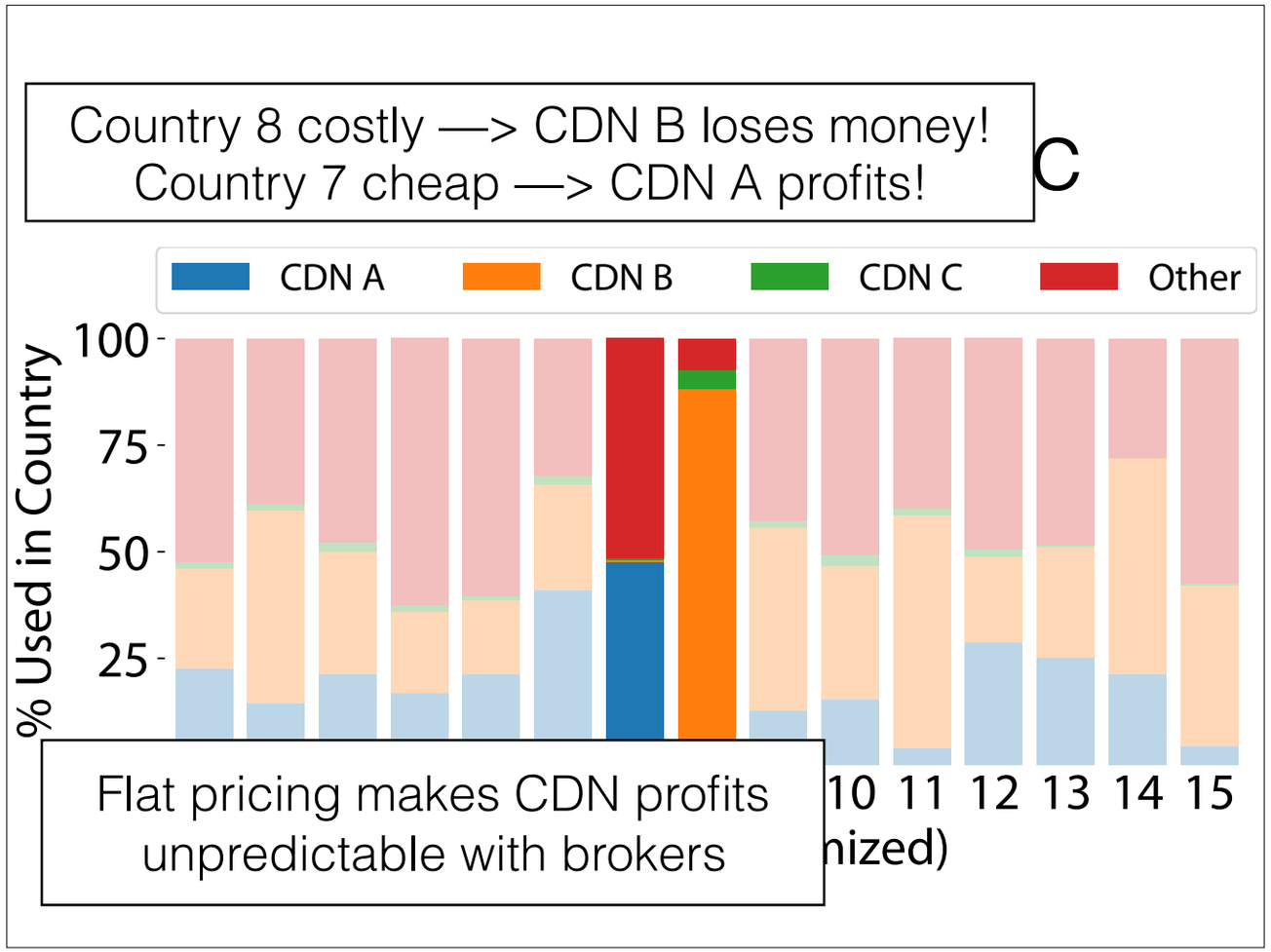
This graphs shows broker data with client requests binned by country. On the x-axis we show the 15 countries with the most requests. The y-axis shows which CDNs served what percentage of clients in each country. Here's the data…



I want to point out two key points of interest: …



… country 8 is predominantly served by CDN B, with few clients served by CDN A. Country 7 is the opposite. Recall that there's a 30 times variation in cost between some countries.



With that in mind, if country 8 is costly, CDN B has difficulty making a profit. If country 7 is cheap, CDN A can easily profit. What this all points to is the larger problem, ** the CDN flat pricing model makes profits unpredictable when traffic is unpredictable (e.g., due to brokers).

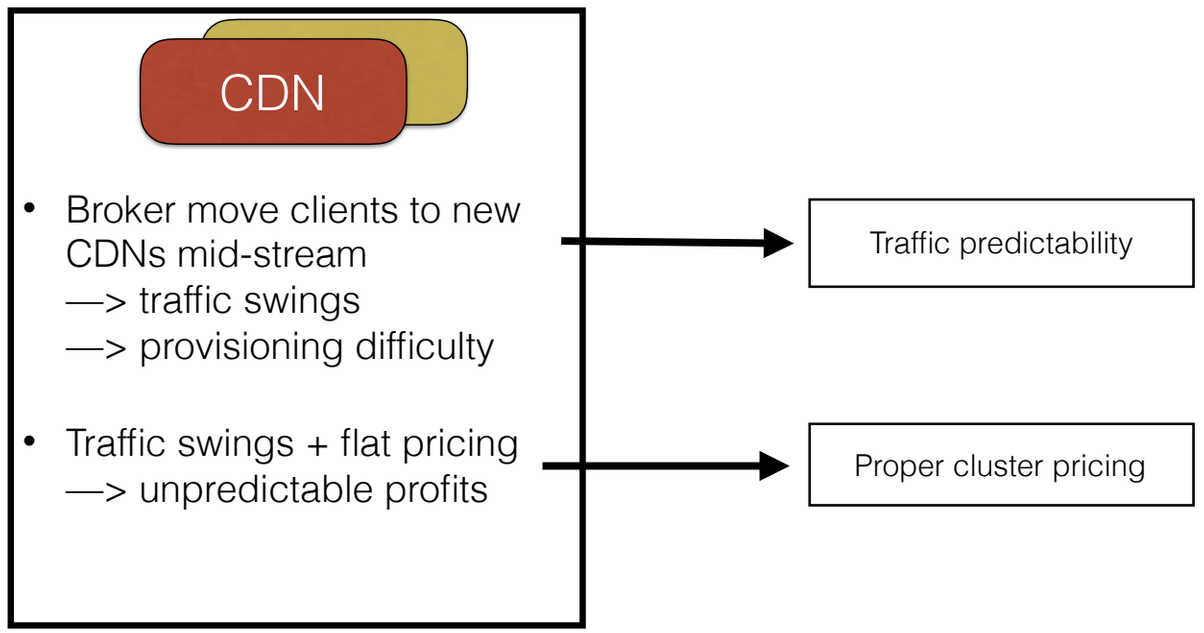
Potential Problems

- Broker move clients to new CDNs mid-stream
  —> traffic swings
  —> provisioning difficulty
- Traffic swings + flat pricing
  —> unpredictable profits

- Coarse CDN-level selection + Incomplete measurements
  —> limited choices
  —> sub-optimal delivery

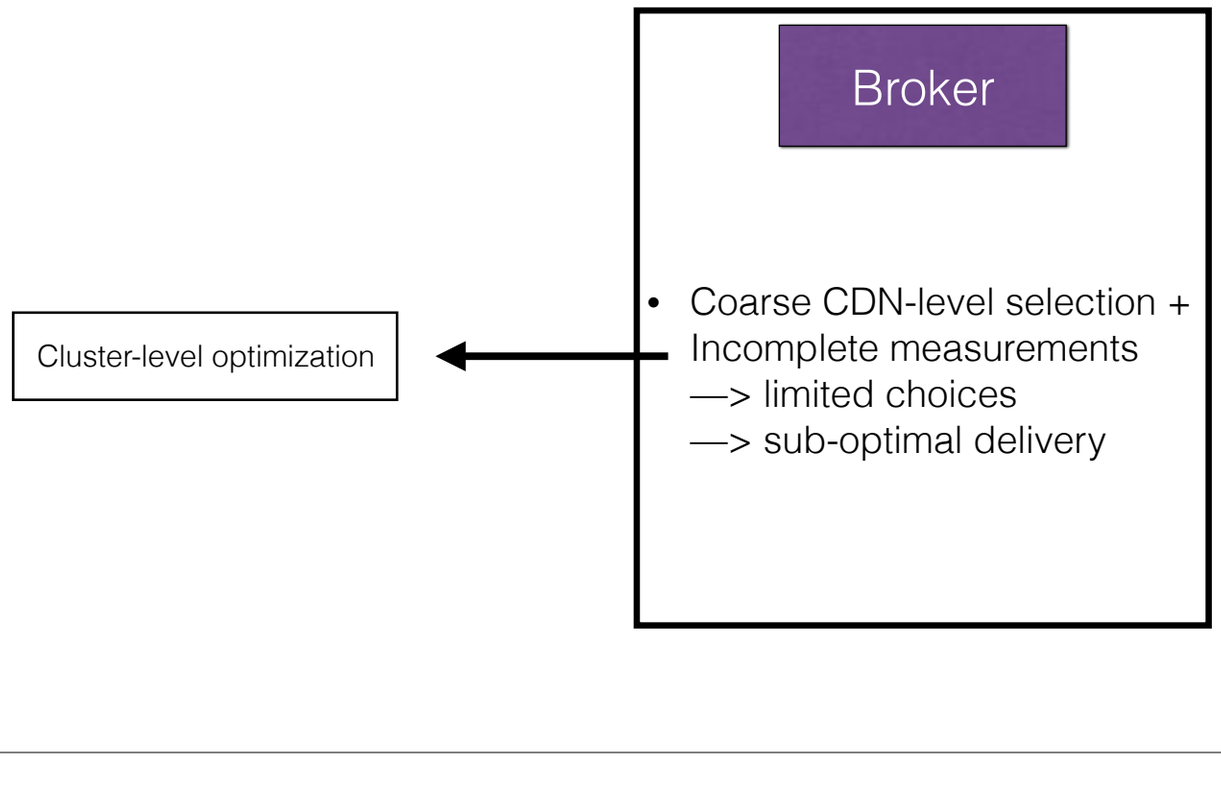
Now that we understand some problems facing CDNs in a world with brokers, let’s talk about how we can fix these problems.

Requirements



Obviously, we can remove broker created traffic swings by ** enforcing some notion of traffic predictability. Unpredictable profits due to traffic swings and flat pricing can be fixed by ** have pricing reflect cluster delivery cost.

Requirements

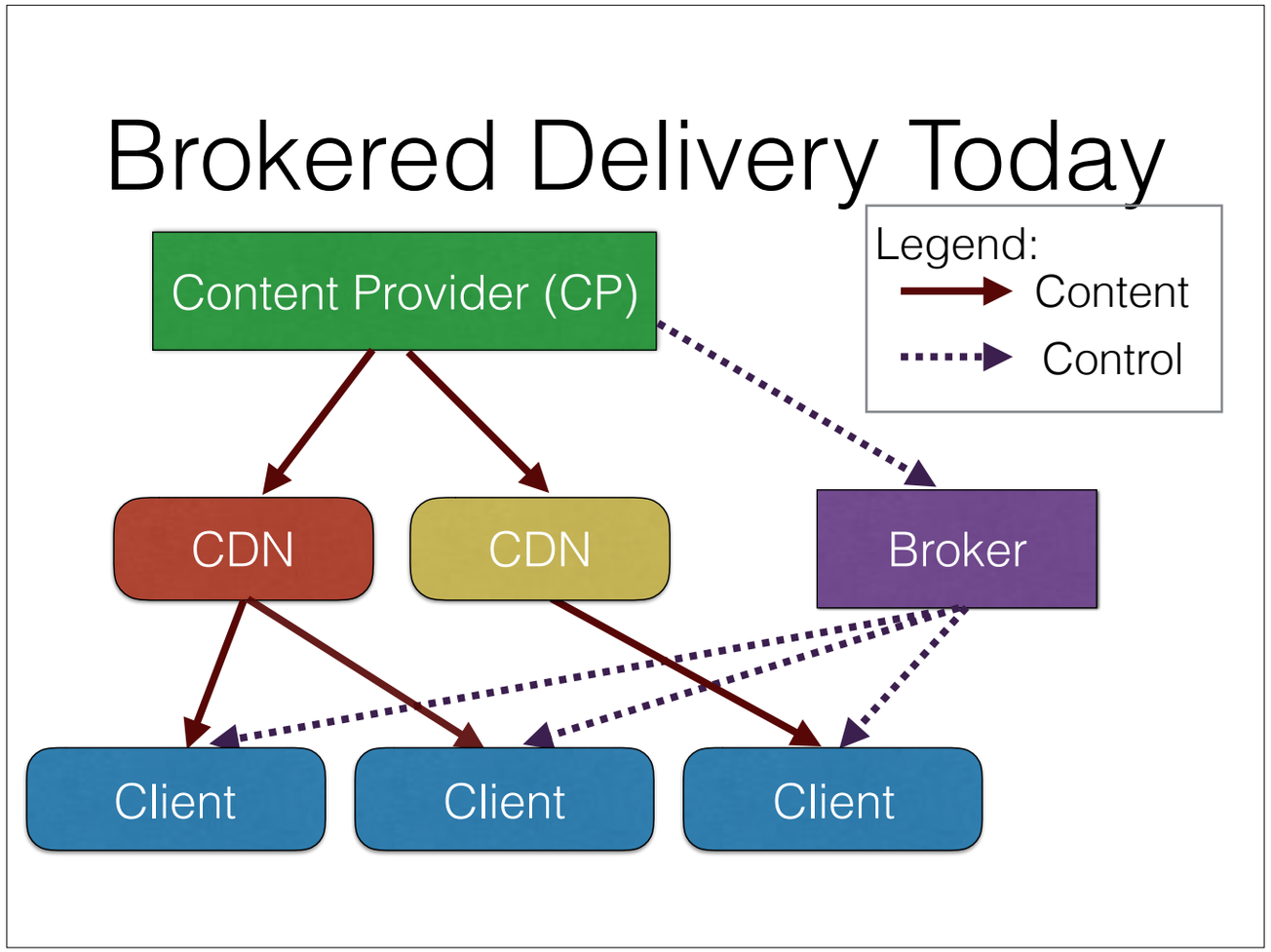


From the broker’s side, limited choices in delivery optimization can be addressed by ** exposing CDN clusters for more fine-grained optimization.

Contributions

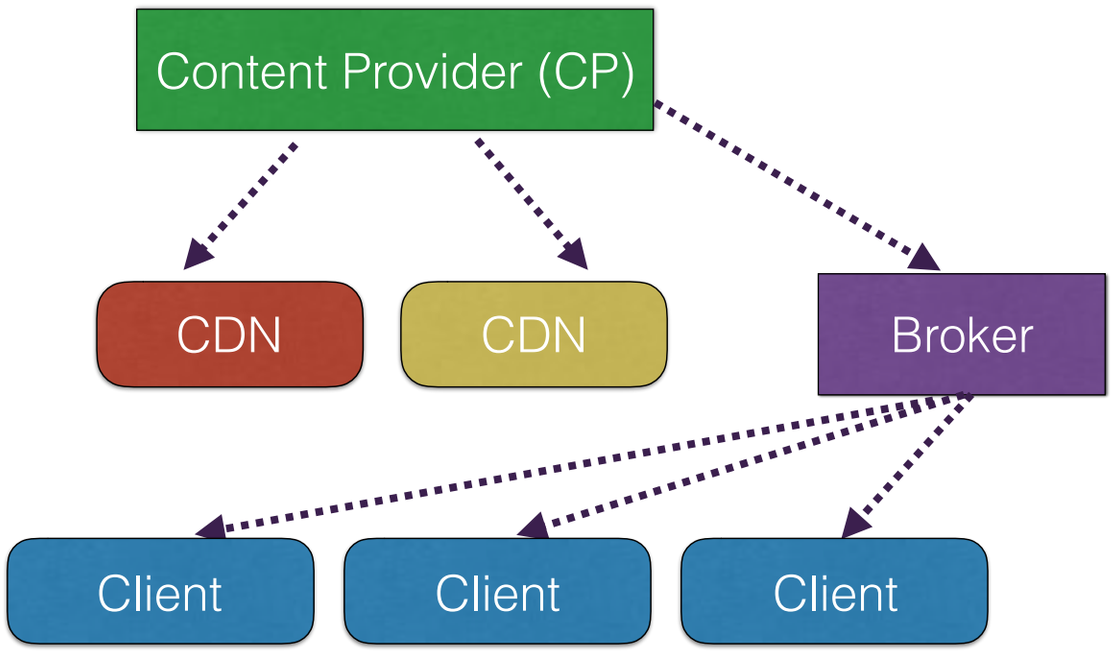
- Identify challenges that brokers and CDNs create for each other by analyzing data from both
- Examine the design space of CDN-broker interfaces
- Evaluate the efficacy of different designs

Now that we understand the requirements of a CDN-broker interface, we need to understand how different designs might meet these requirements. To do that we need to first understand today's brokered content delivery control plane in more detail.



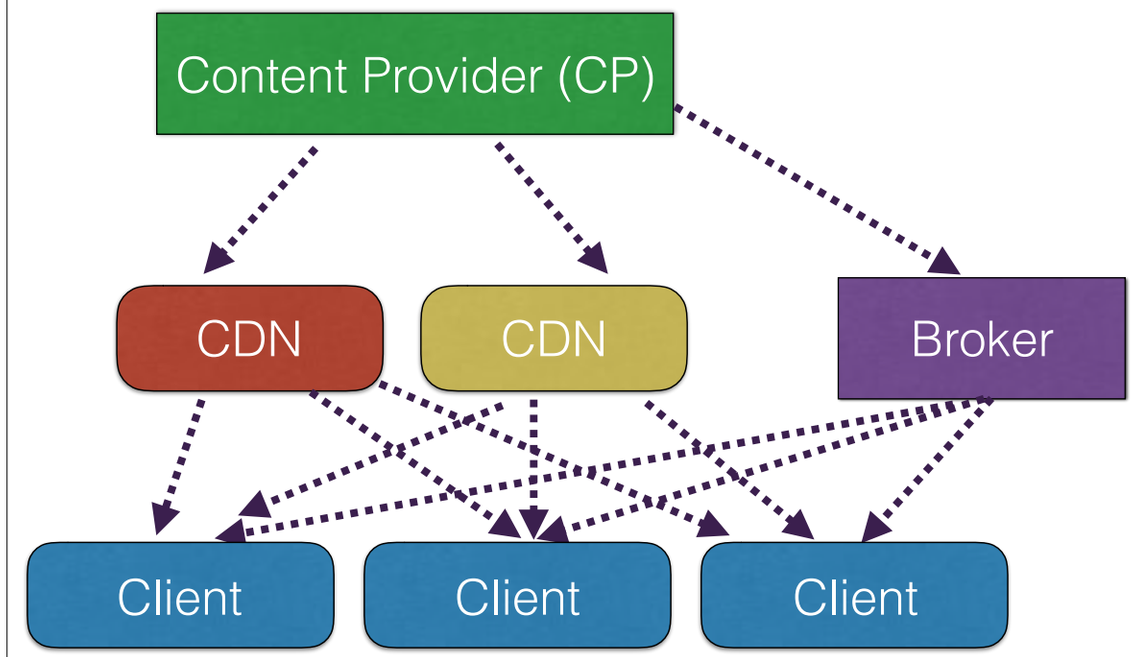
Here the picture we saw earlier of brokered content delivery today. Even with the addition of brokers, the actual real-time aspects of content delivery is still the same, so let’s focus just on the control plane **. It’s not quite complete though…

Brokered Delivery Today



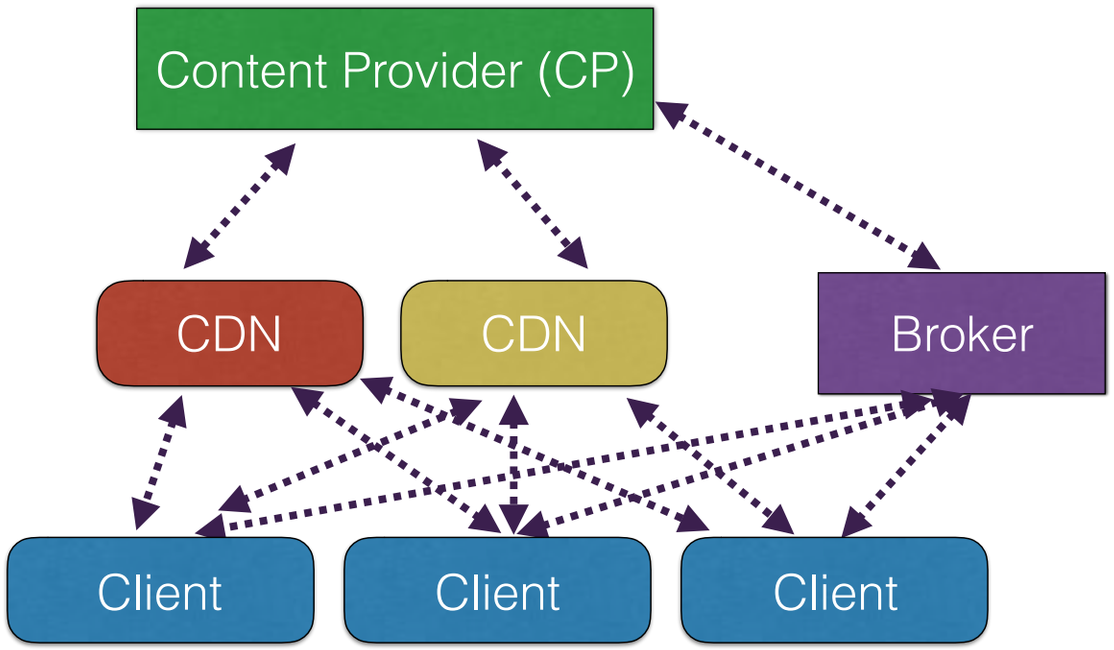
CPs pass control information to CDNs (e.g., what content they’re allowed to serve), and CDNs…

Brokered Delivery Today



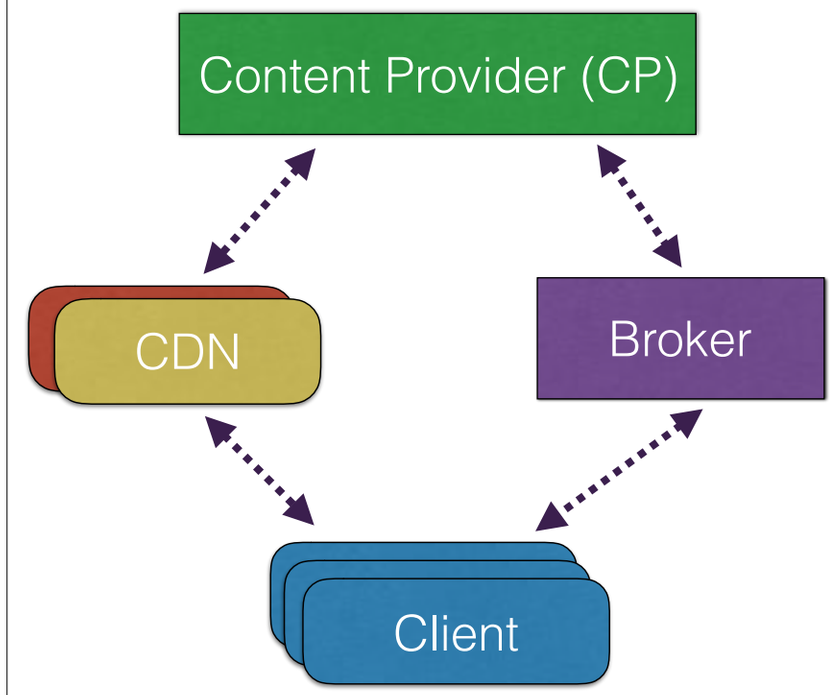
… map clients to specific clusters within the CDN. Information flow isn’t simply one way though (e.g., CDNs gather performance estimates from clients). So the picture looks more like this…

Brokered Delivery Today



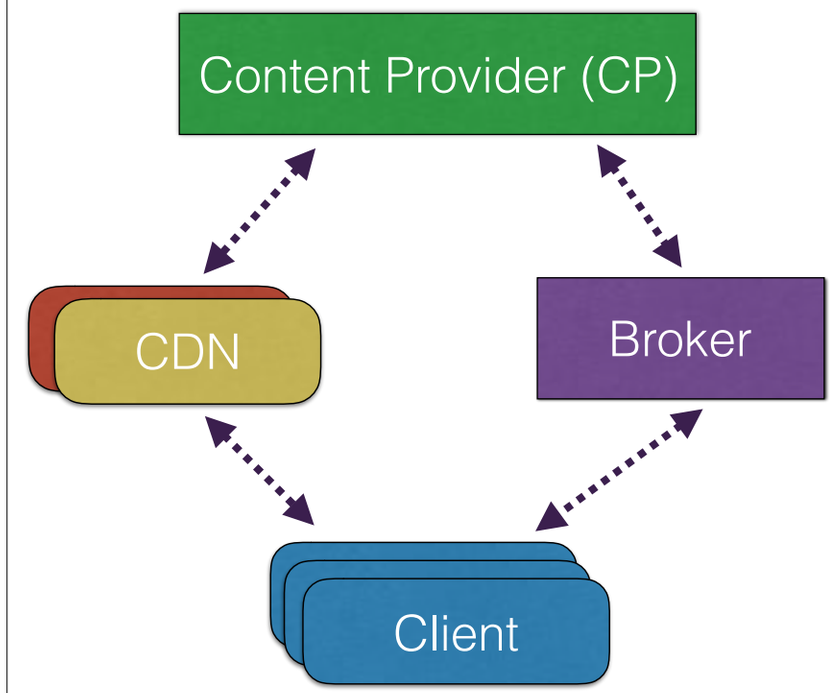
With double-headed arrows. Okay, this is getting a bit cluttered, so let’s simplify it.

Brokered Delivery Today



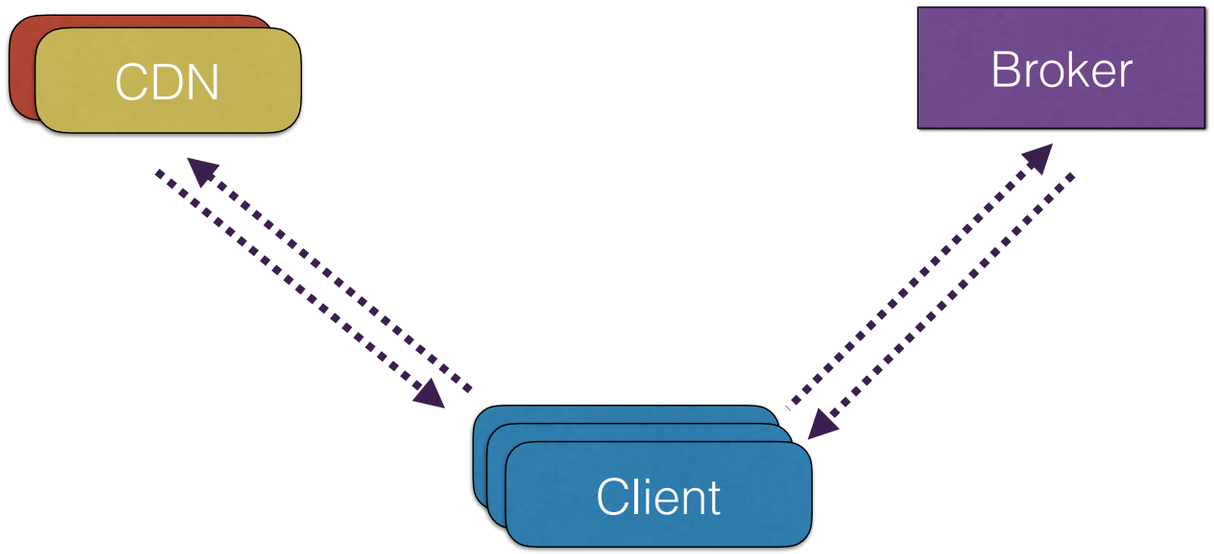
There, that’s better. CP interactions (contract negotiations) are at a much longer timescale (e.g., months/years) so let’s further simplify things by removing them.

Brokered Delivery Today



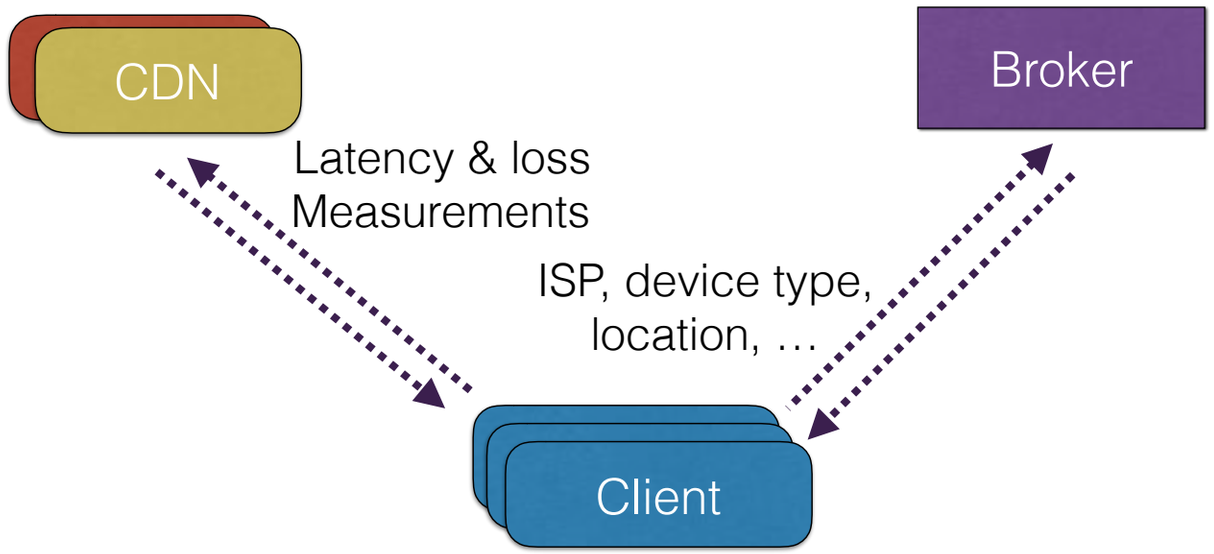
There, that’s better. Content provider interactions (i.e., contract negotiations) are at a much longer timescale today (e.g., months/years) so let’s further simplify things by removing them.

Brokered Delivery Today



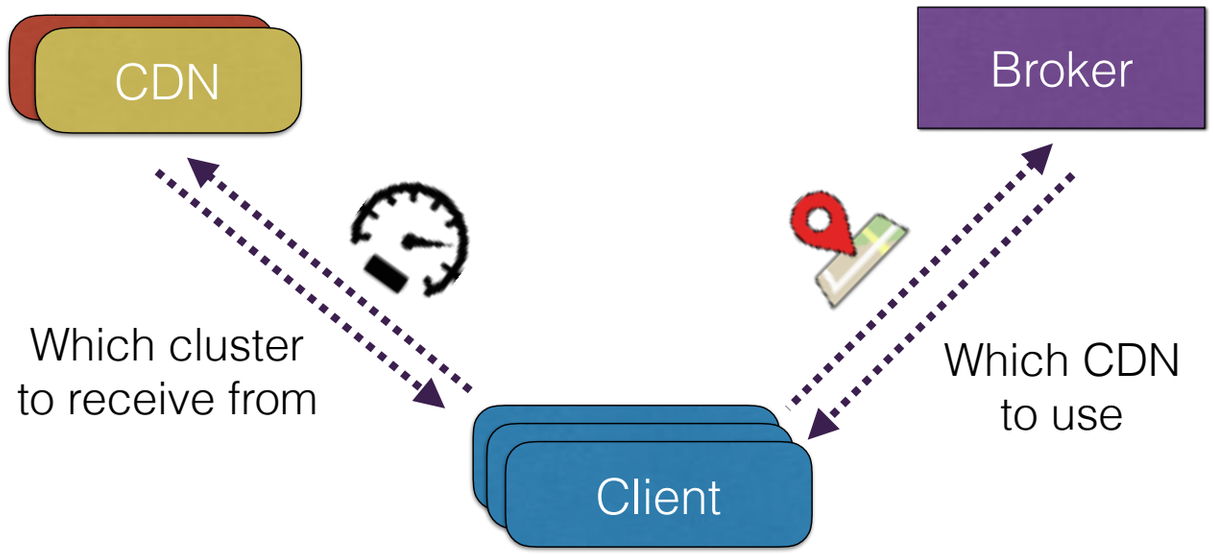
Okay. Let’s be more concrete. What information is shared between CDNs and clients and the broker and clients today? ** Clients provide CDNs with latency and loss measurement (i.e., network performance). ** Clients provide brokers with meta-data about the client (e.g., ISP, device type, geographic location, etc.) Getting a little cluttered so let’s make these into icons…

Brokered Delivery Today



Okay. Let’s be more concrete. What information is shared between CDNs and clients and the broker and clients today? ** Clients provide CDNs with latency and loss measurement (i.e., network performance). ** Clients provide brokers with meta-data about the client (e.g., ISP, device type, geographic location, etc.) Getting a little cluttered so let’s make these into icons…

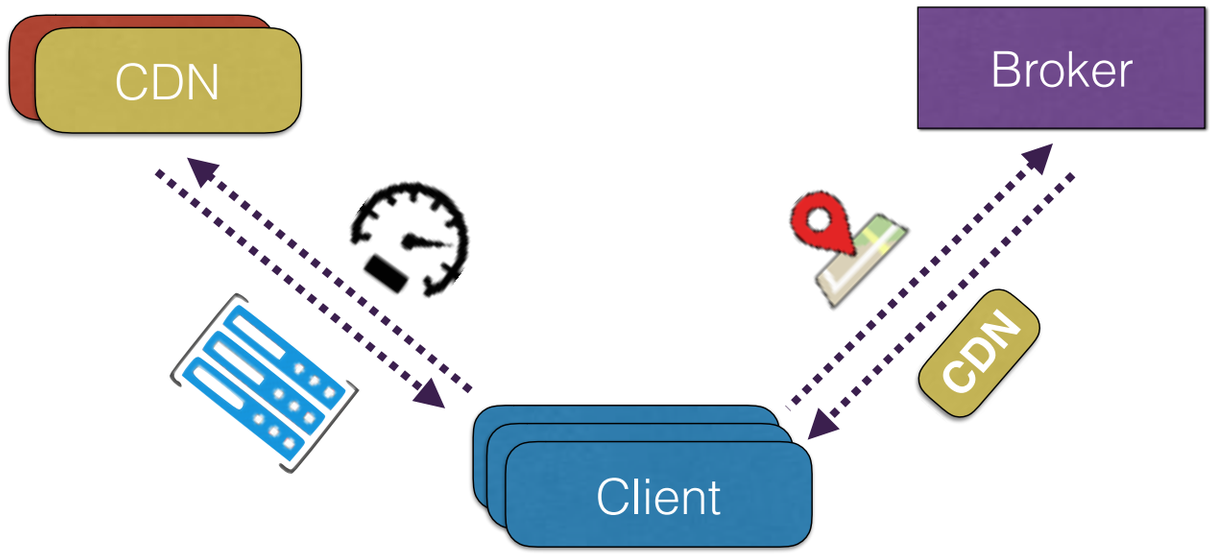
Brokered Delivery Today



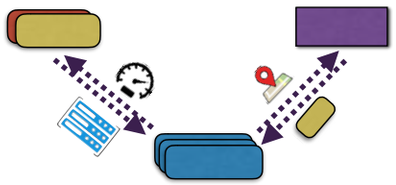
So, network measurements and meta-data like location.

CDNs tell clients ** which of their clusters to go to, and the broker tells clients ** which CDN to use. More cleanly…

Brokered Delivery Today



A cluster and a CDN. Does this design meet our requirements? Well…



Today

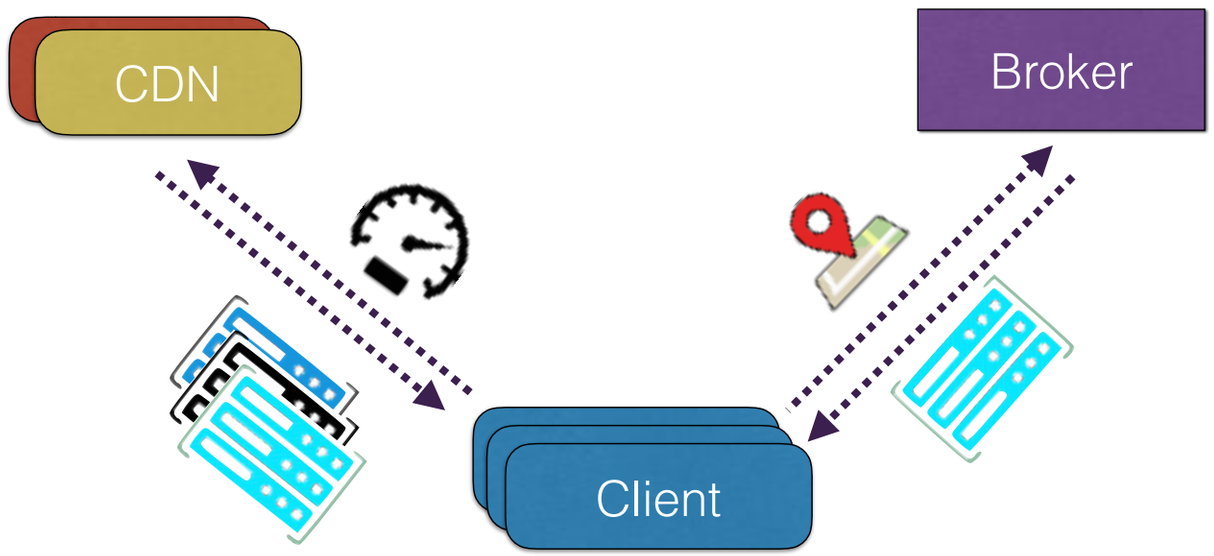
❌ predictable

❌ price shows cost

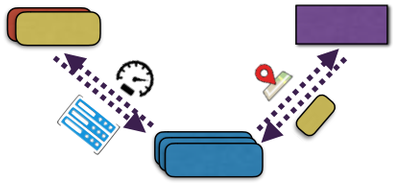
❌ cluster level opt.

Looking back to them, with brokering today, ** brokers move clients around arbitrarily (thus no traffic predictability), ** cluster-level costs aren’t communicated anywhere (so prices don’t reflect cost), ** and brokers optimize over whole CDNs (not clusters).

Multicluster

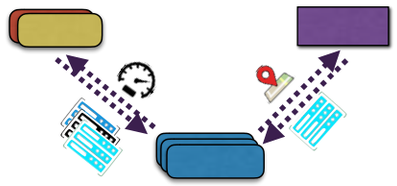


Clearly we need to make some changes. You might notice that there are no arrows between CDNs and the broker as there isn’t an interface there today. But this is only one place we can change. A more simple fix to get finer-grained optimization would be to have CDNs allow clients or brokers to select between multiple clusters ** **. That means the broker can make decisions at the cluster level, not the CDN level ** to meet more nuanced performance and cost goals.



Today

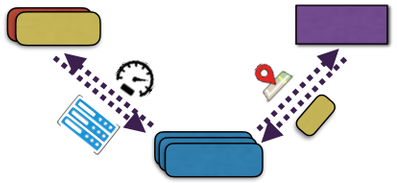
❌ predictable
❌ price shows cost
❌ cluster level opt.



Multicluster

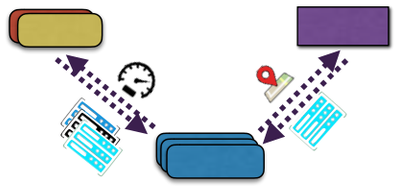
❌ predictable
❌ price shows cost
✅ cluster level opt.

We call this approach “Multicluster”. We clearly meet the cluster-level optimization requirement.



Today

❌ predictable
❌ price shows cost
❌ cluster level opt.

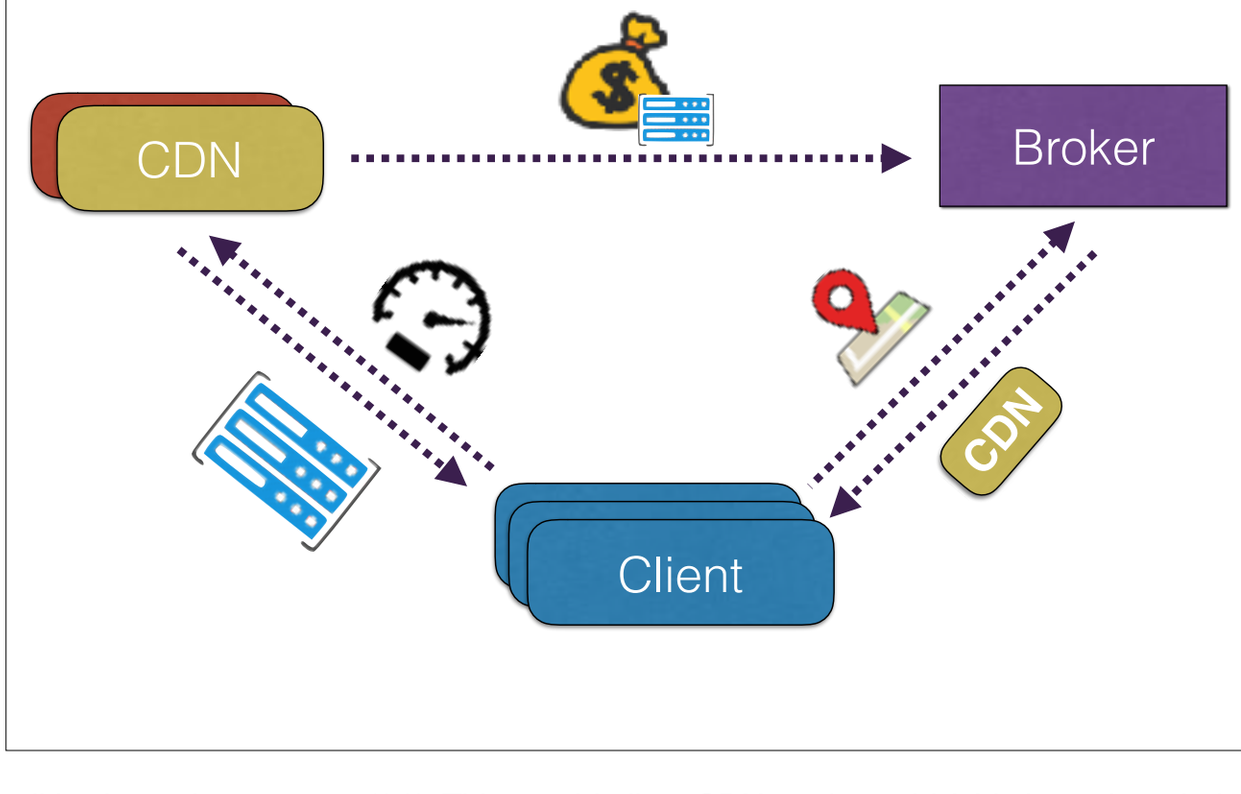


Multicluster

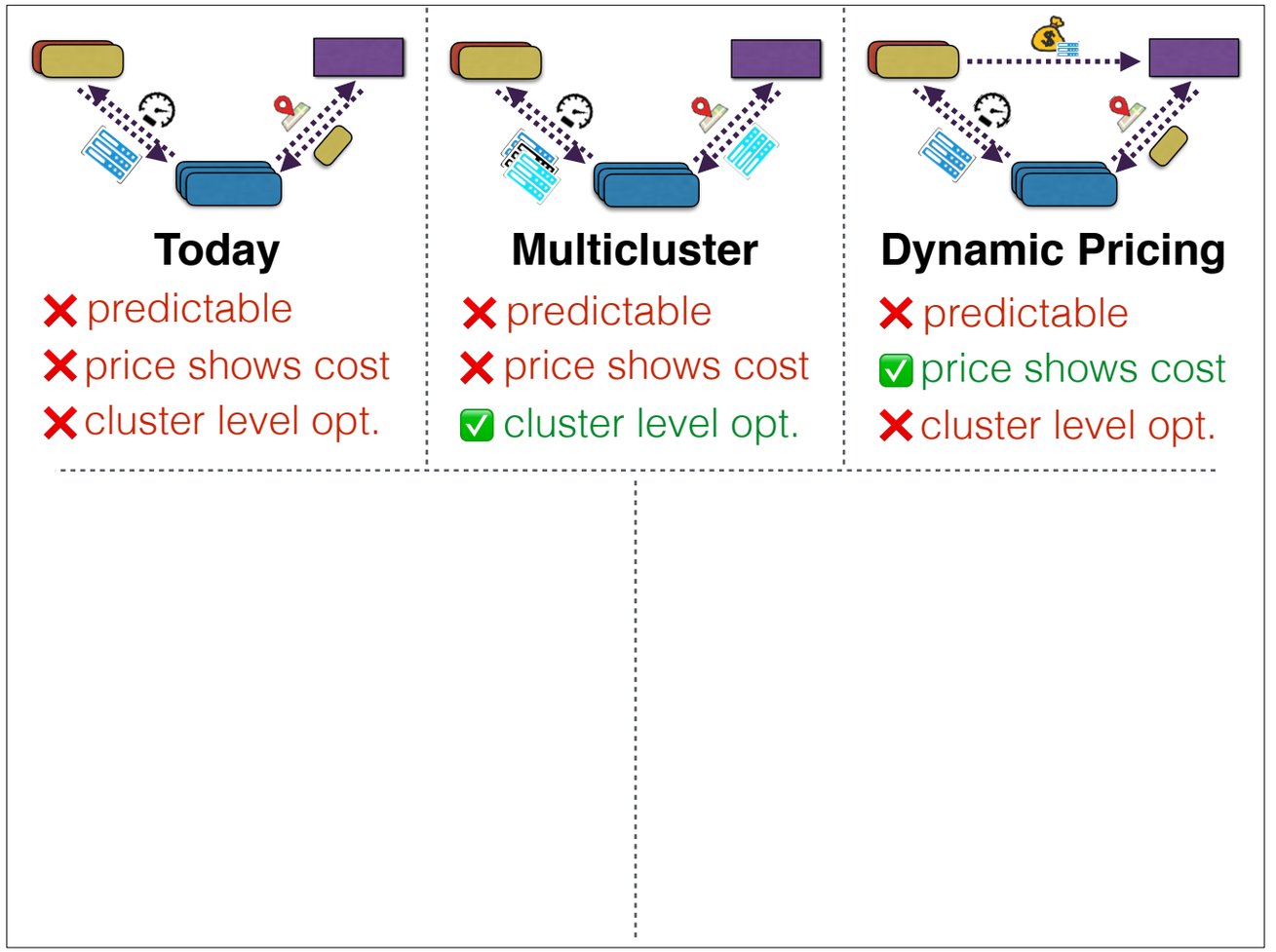
❌ predictable
❌ price shows cost
✅ cluster level opt.

We call this approach “Multicluster”. We clearly meet the cluster-level optimization requirement.

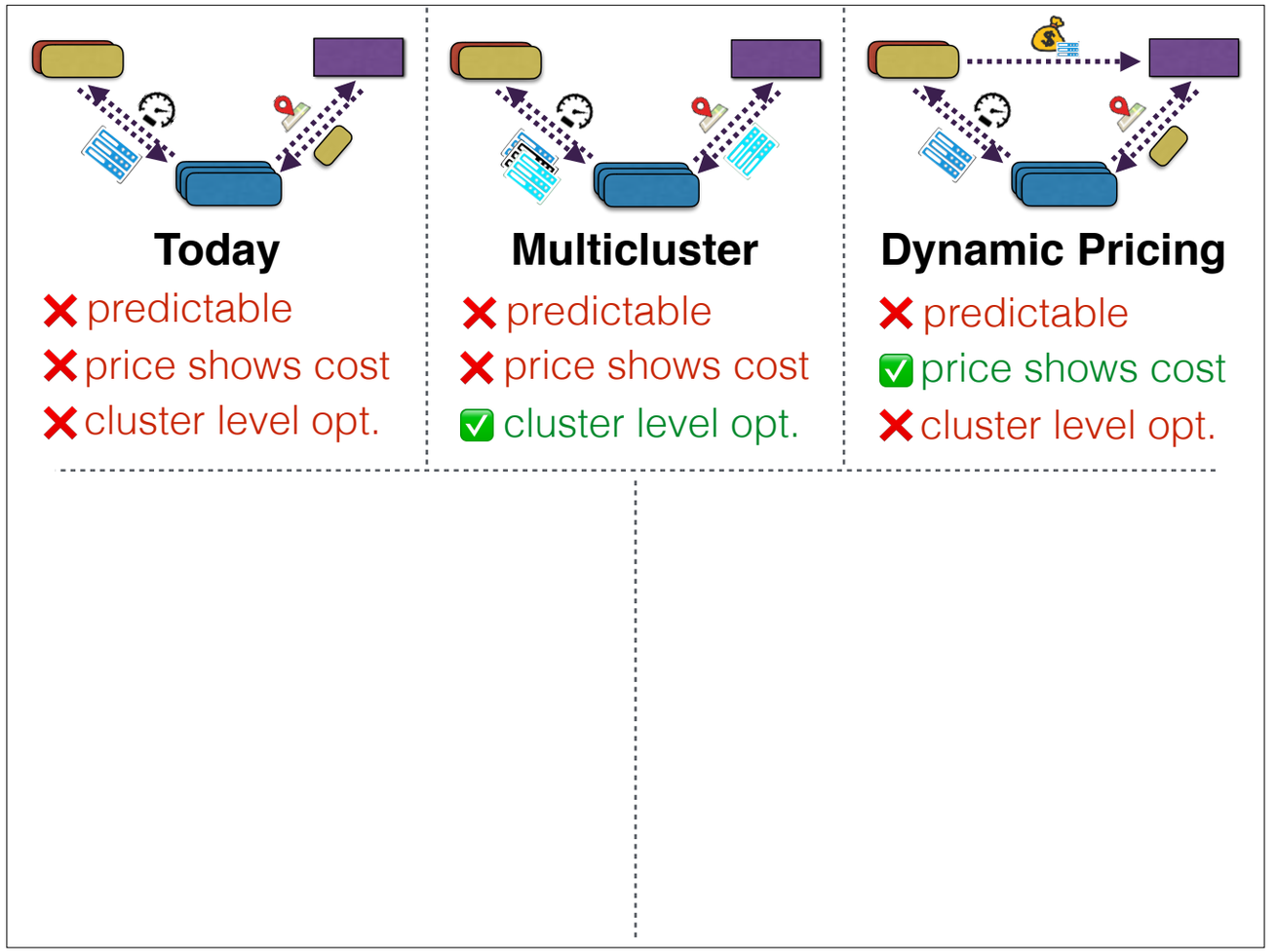
Dynamic Pricing



Another simple fix would be have CDNs tell brokers cluster costs ** **. This would allow CDNs to be paid fairly based on their internal costs.

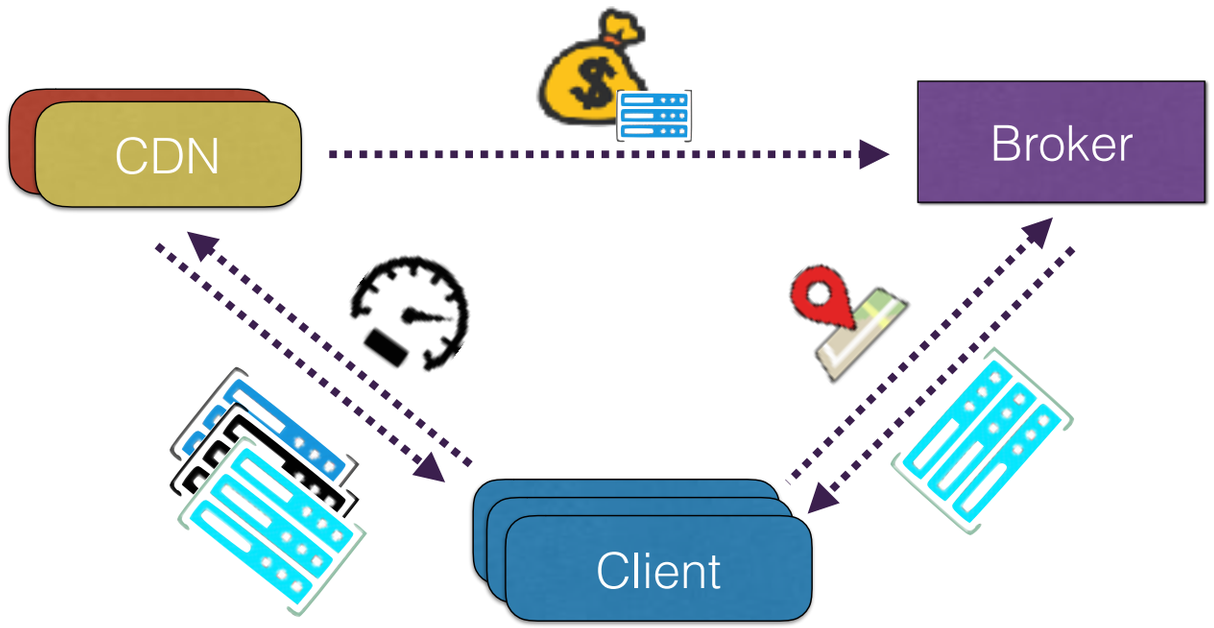


We call this approach “Dynamic Pricing”. We clearly meet the ‘price shows cost’ requirement.

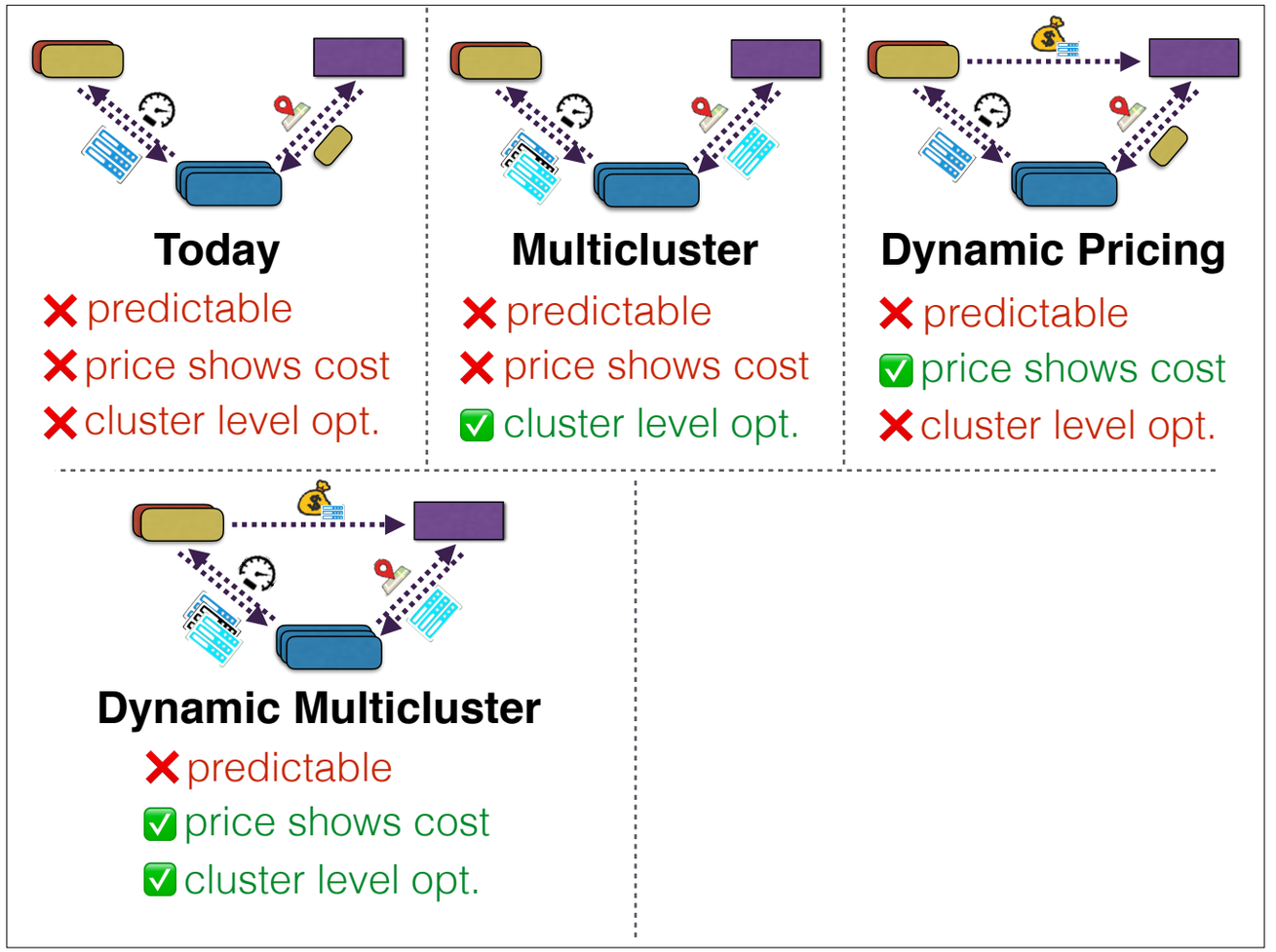


We call this approach “Dynamic Pricing”. We clearly meet the ‘price shows cost’ requirement.

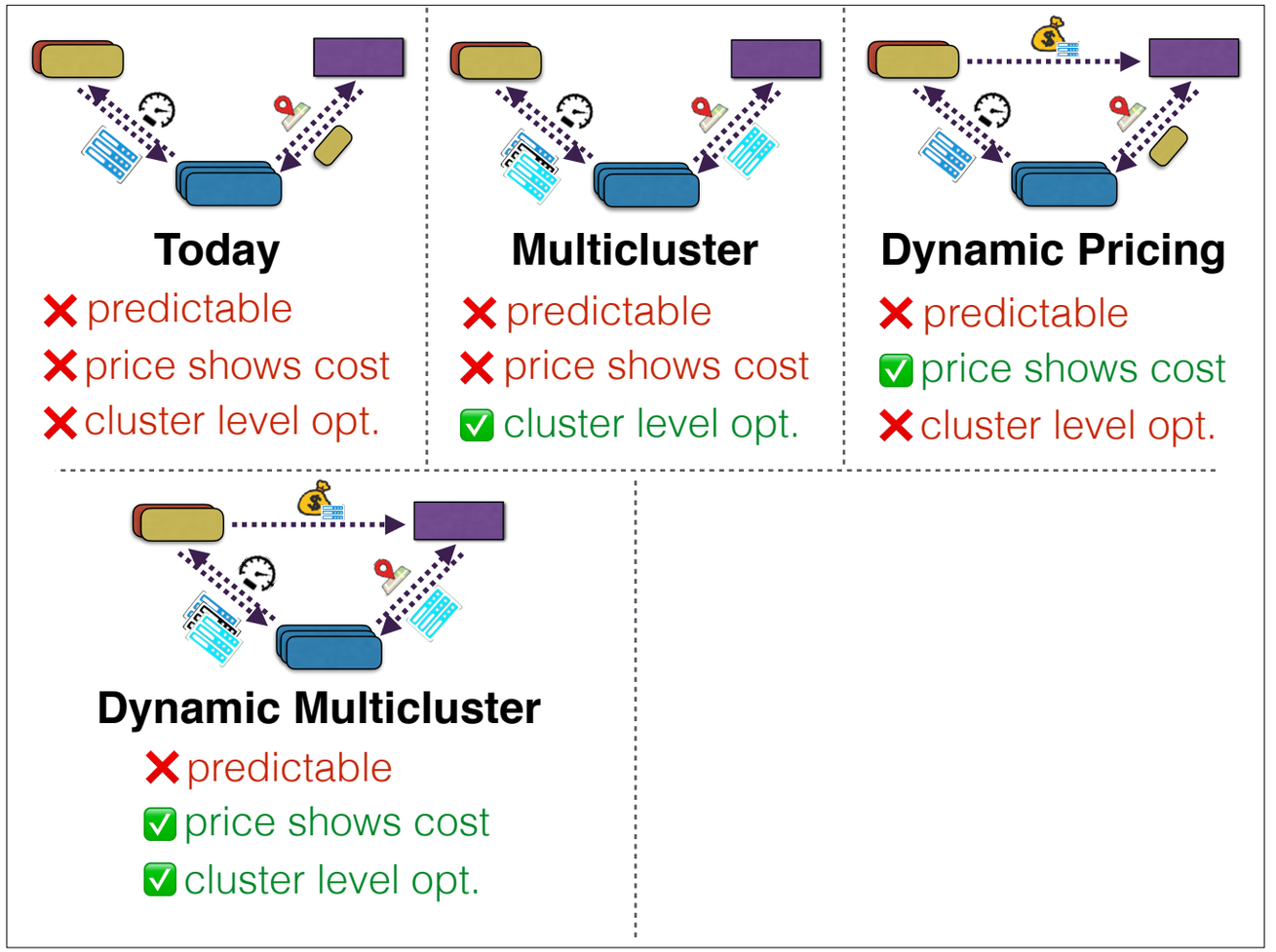
Dynamic Multicluster



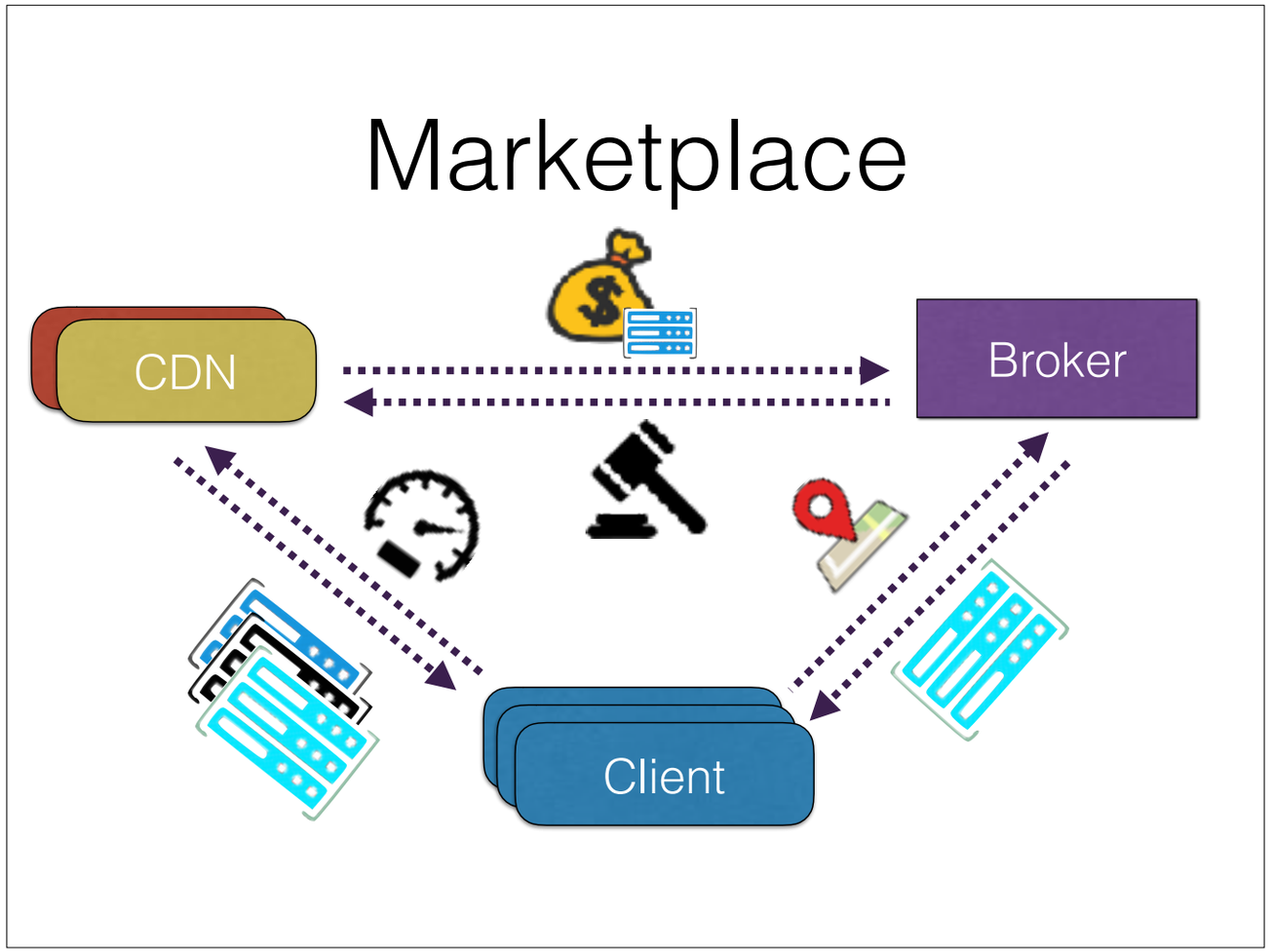
We could also just do both… ** ** multiple clusters **, and ** ** dynamic pricing.



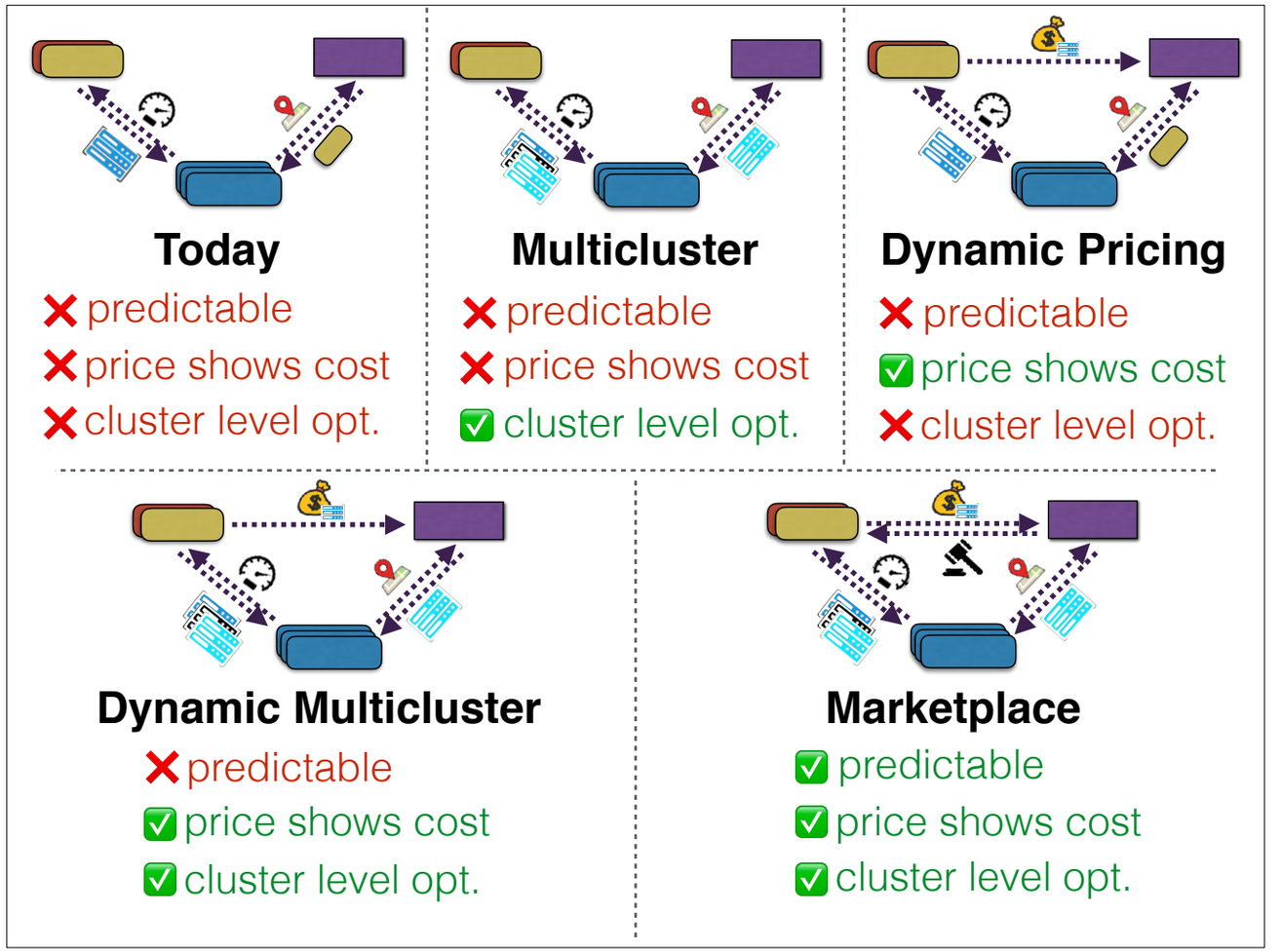
We call this approach “Dynamic Multicluster”. It provides cluster-level optimization and proper pricing, but not traffic predictability. Why is that?



We call this approach “Dynamic Multicluster”. It provides cluster-level optimization and proper pricing, but not traffic predictability. Why is that?



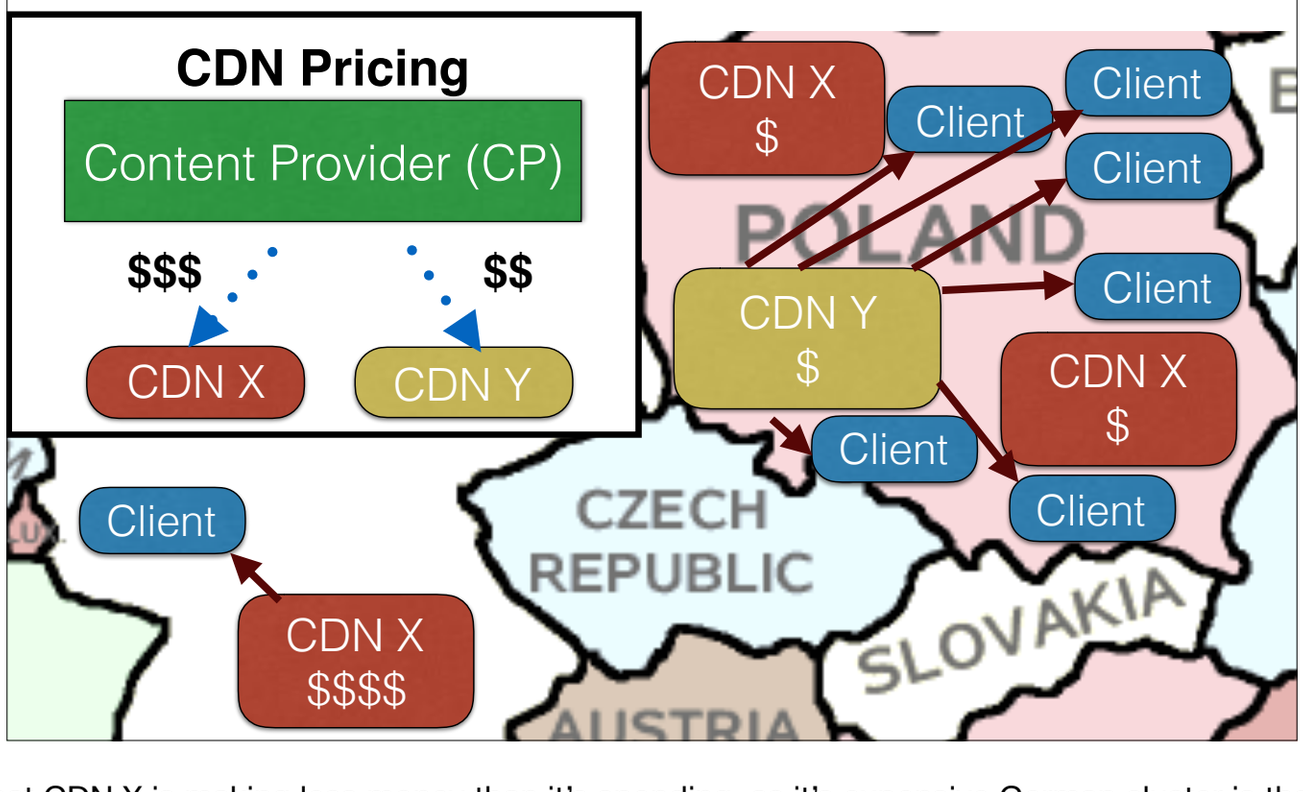
The problem is that the broker doesn’t tell CDNs when they’re going to be moving large groups of clients, causing load balancing problems for CDNs. We could fix this by simply having ** ** brokers announce their choices to CDNs. This starts to look like a “Marketplace”, where CDNs are ‘bidding’ the broker to use their clusters.



Since the results of the marketplace are announced by the broker before clients are moved, we get much more traffic predictability.

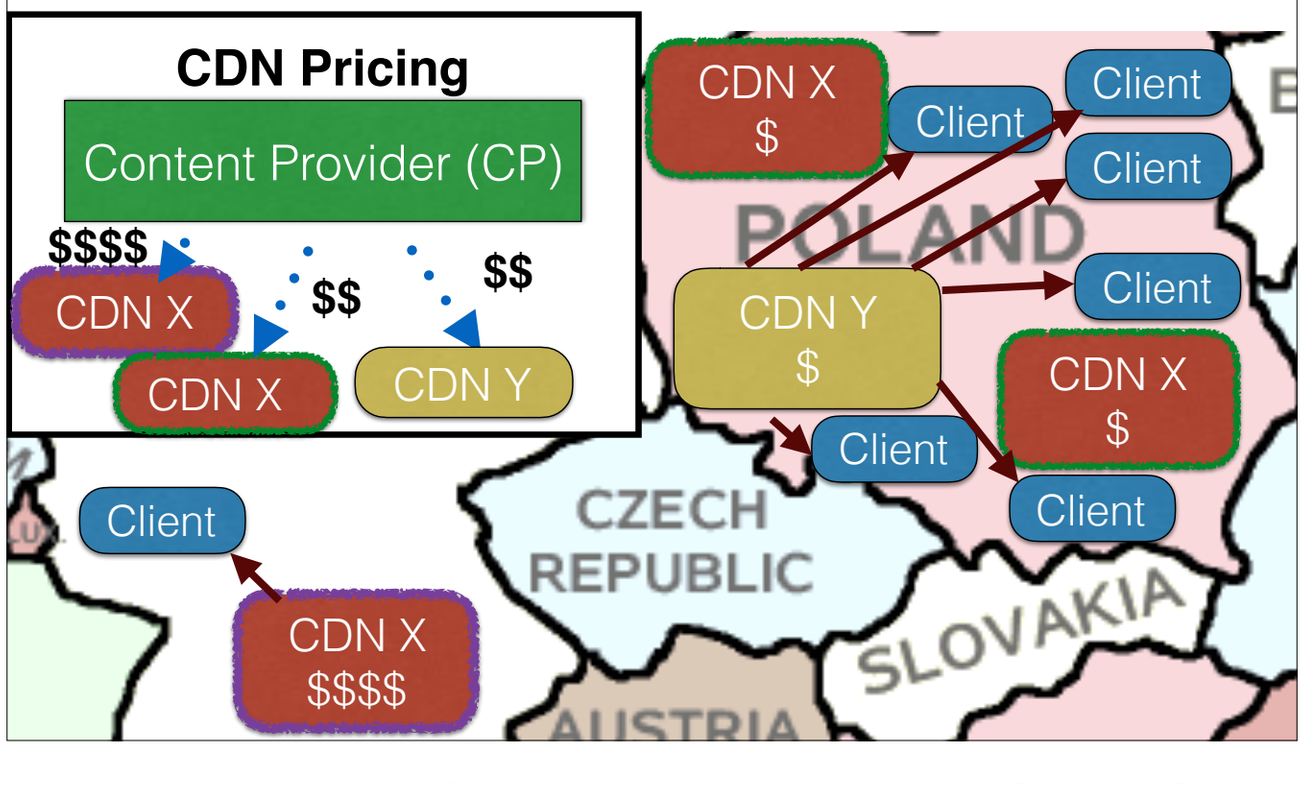
Let’s go back to our previous example and see how these systems can solve their problems.

Example



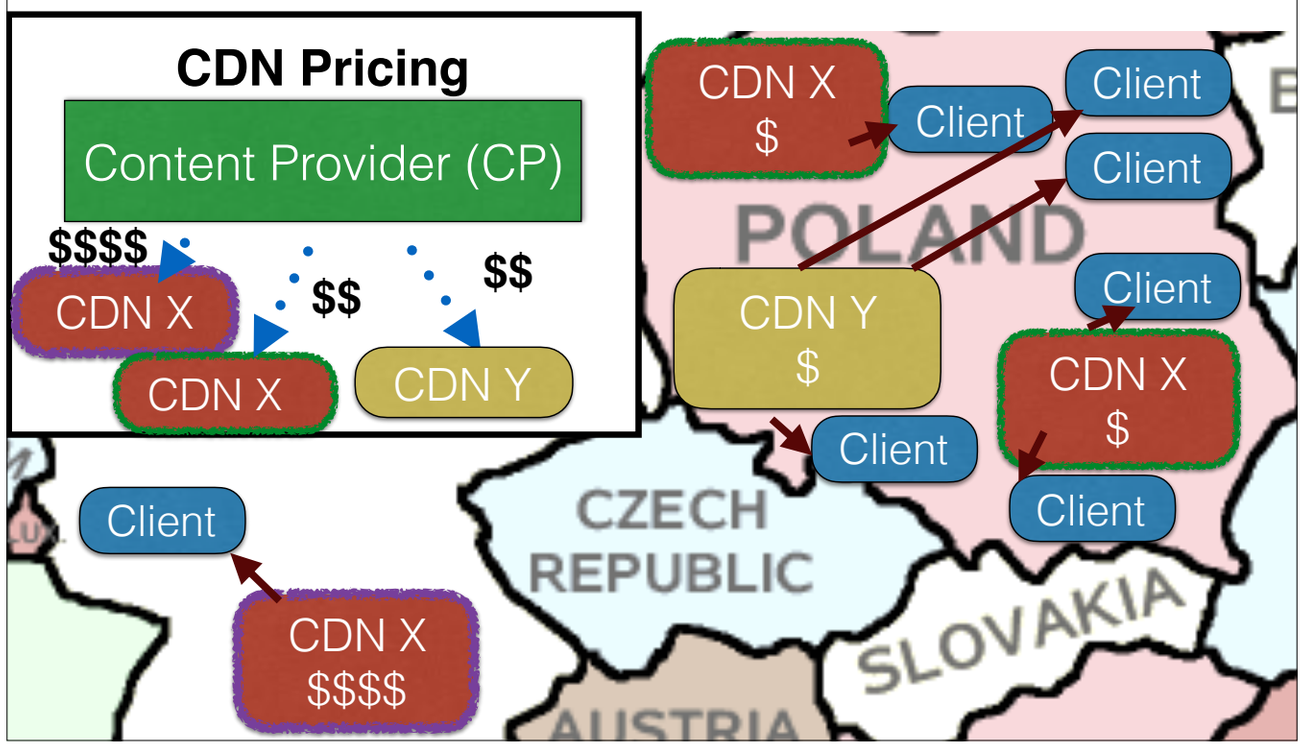
Recall this example. The problem was that CDN X is making less money than it's spending, as it's expensive German cluster is the only one used by the broker. With some our designs, individual clusters within the CDN can have different prices reflecting their delivery cost. Let's fix this example by splitting CDN X into two different groups, Germany and Poland.

Example



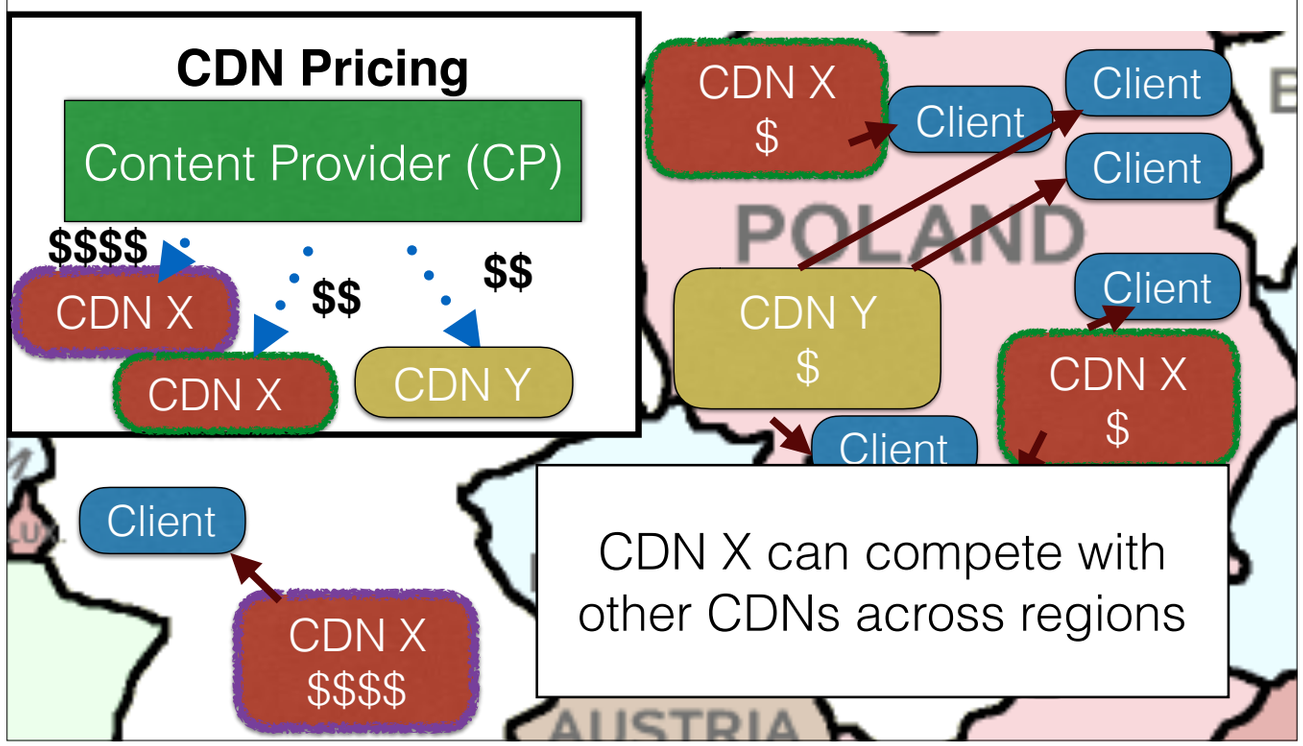
Now, CDN X can price their German cluster at cost, while pricing their Poland clusters to be competitive with CDN Y. As CDN X is now priced competitively in Poland, the broker …

Example



… may move some traffic in Poland to CDN X, …

Example



… allowing CDN X to compete with other CDNs across regions.

Contributions

- Identify challenges that brokers and CDNs create for each other by analyzing data from both
- Examine the design space of CDN-broker interfaces
- Evaluate the efficacy of different designs

Now let’s evaluate how effective each design is.

Evaluation

- Simulator using data from a broker & CDN, as well as public data from 13 other CDNs
- CDN data provides cluster locations, cluster-to-client performance, delivery costs, etc.
- Broker data provides client locations, request distributions, etc.

We build a simulator using data from a broker and data from a CDN as well as public data from 13 other CDNs. The CDN data provides cluster locations, cluster-to-client performance estimates, delivery costs, etc. The broker data provides client locations, request distributions, etc.

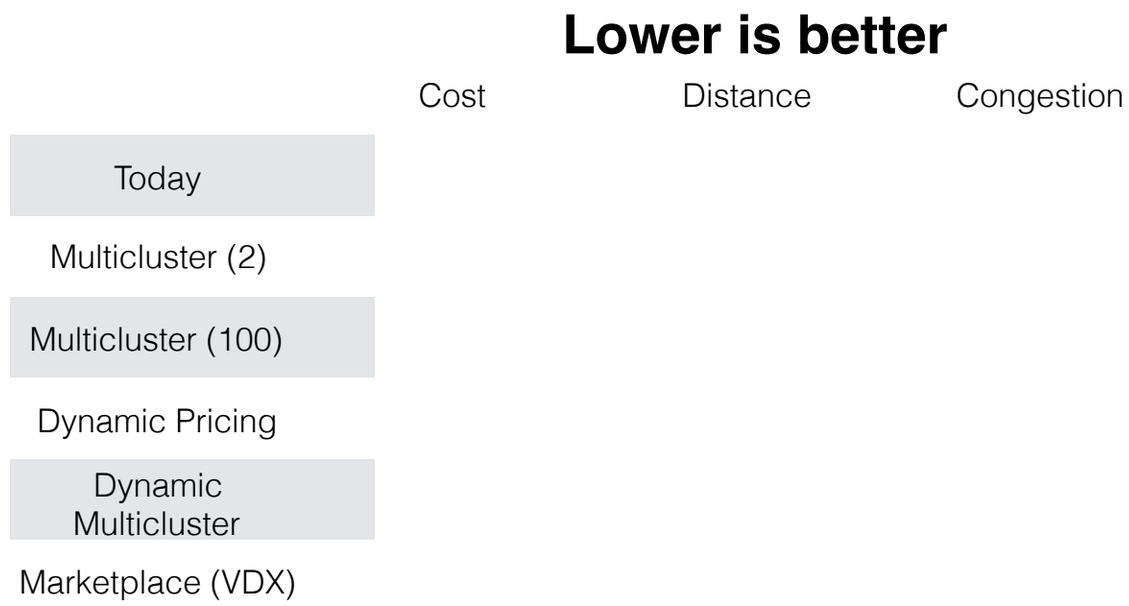
Evaluating the Designs

Lower is better

Cost Distance Congestion

Let’s compare the designs we’ve previously seen in our simulator to find which design provides the most promise. We’re going to compare based on three metrics: CDN internal cost, client-to-cluster distance (a proxy for performance), and congestion (which is the percentage of CDN clusters that are overloaded). For all three metrics, lower numbers are better.

Evaluating the Designs



Here are the designs we looked at: Brokered delivery today, multicluster (exposing 2 and 100 clusters respective), dynamic pricing, dynamic multicluster, and a marketplace design. We call our implementation of a marketplace VDX (for “Video delivery exchange”).

Evaluating the Designs

Lower is better

	Cost	Distance	Congestion
Today	136	297	
Multicluster (2)	155	194	
Multicluster (100)	171	141	
Dynamic Pricing			
Dynamic Multicluster			
Marketplace (VDX)			

First, we see that adding multiple clusters ** decreases distance (i.e., providing better performance) and that exposing more clusters ** (going from 2 to 100) provides even better performance.

Evaluating the Designs

Lower is better

	Cost	Distance	Congestion
Today	136	297	
Multicluster (2)	155	194	
Multicluster (100)	171	141	
Dynamic Pricing	126	318	
Dynamic Multicluster			
Marketplace (VDX)			

Next, we see that having the CDN price reflect the internal cluster delivery cost ** does lead to ** lower overall delivery cost as the broker becomes more cost aware.

Evaluating the Designs

Lower is better

	Cost	Distance	Congestion
Today	136	297	
Multicluster (2)	155	194	
Multicluster (100)	171	141	
Dynamic Pricing	126	318	
Dynamic Multicluster	115	219	
Marketplace (VDX)			

Next we see that exposing more clusters and their costs gives us the ** best of both worlds — lower cost and lower distance.

Evaluating the Designs

Lower is better

	Cost	Distance	Congestion
Today	136	297	0%
Multicluster (2)	155	194	27%
Multicluster (100)	171	141	39%
Dynamic Pricing	126	318	0%
Dynamic Multicluster	115	219	14%
Marketplace (VDX)			

This comes at the cost of congestion though.

Evaluating the Designs

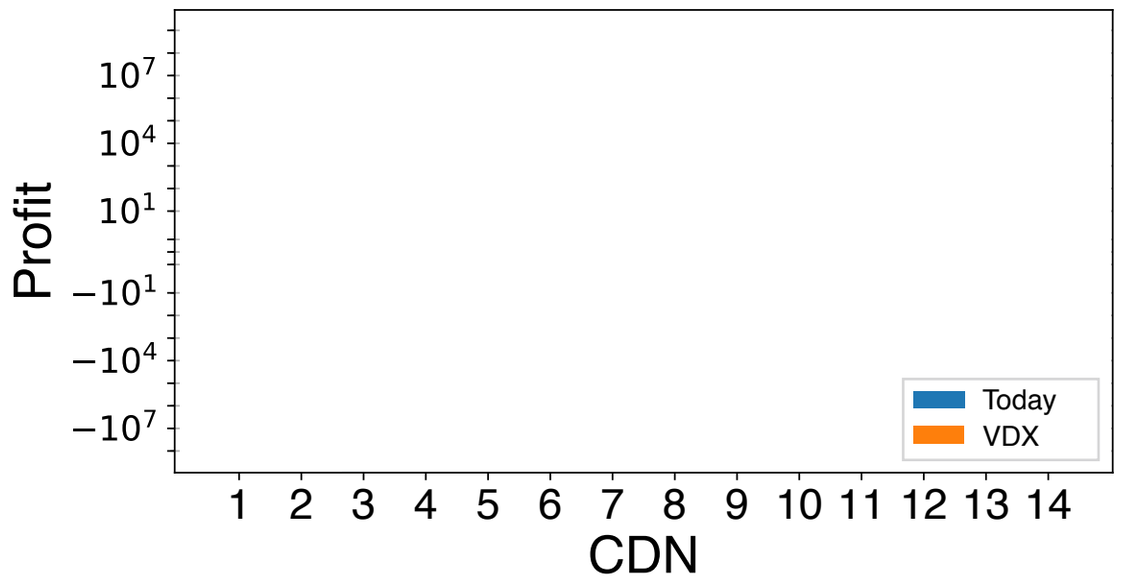
Lower is better

	Cost	Distance	Congestion
Today	136	297	0%
Multicluster (2)	155	194	27%
Multicluster (100)	171	141	39%
Dynamic Pricing	126	318	0%
Dynamic Multicluster	115	219	14%
Marketplace (VDX)	93	178	0%

Most promising

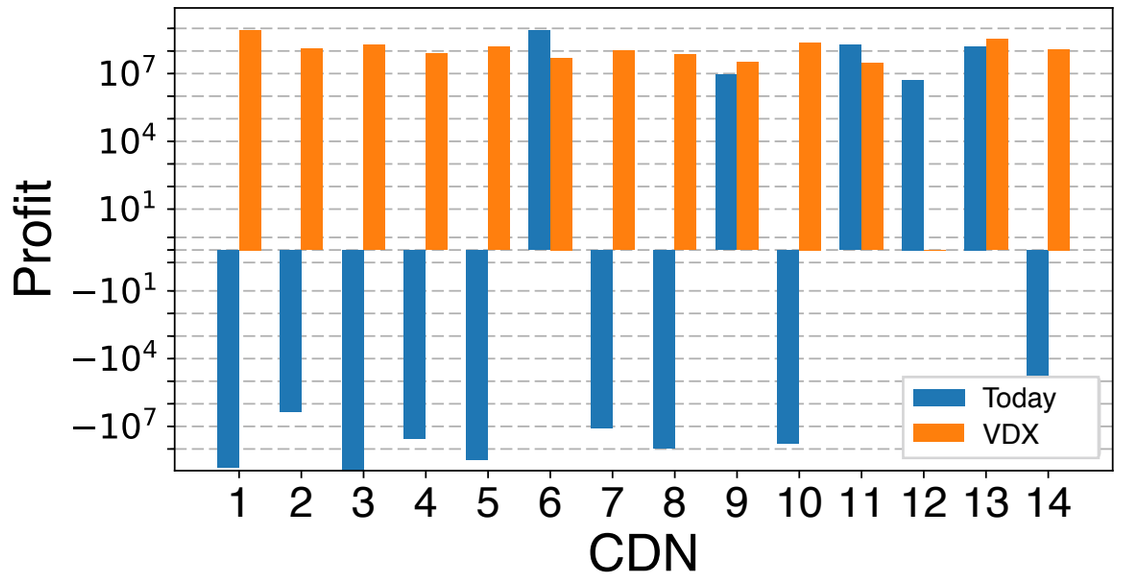
Finally, our marketplace design VDX provides the ** overall lowest cost, with ** significantly improved performance over today, although it is not the best performing. It does so, ** without causing any congestion. Thus, we believe a marketplace design is the ** most promising.

Per-CDN Profits



I want to show one other taste of the evaluation before wrapping up. Here we're looking at a graph of per-CDN profits in our simulator. The x-axis shows the 14 different CDNs we looked at. The y-axis shows their profits (i.e., how much the CDNs charge content providers relative to their internal delivery cost). We're going to compare brokered video delivery today to our marketplace design VDX.

Per-CDN Profits



What we find is that most CDNs in today's world don't make a profit on brokered video delivery in our simulator. This makes sense given some public quarterly earning reports and that anecdotally CDNs generally don't consider video delivery profitable. There haven't been major alarms raised about the unprofitability of brokered video delivery, as brokered delivery also currently only makes up a small (but growing) portion of overall video delivery, so we expect these problems to become more prominent in the future. VDX on the other hand allows all CDNs to profit because their prices more closely reflect their internal costs.

Other Results

- Deep dive into per-CDN and per-country results
- Adding hundreds of “single-city” CDNs
- Tuning VDX’s performance / cost tradeoff

Other results in the paper include: a deep dive into per-CDN and per-country results, a scenario adding hundreds of “single-city” CDNs to our trace, and tuning VDX’s performance / cost tradeoff. If you’re interested, please read the paper.

Evaluation Takeaways

- Today's world (Brokered) is pretty broken (performance can be better; most CDNs lose money on brokered video delivery)
- Marketplace (VDX) fixes this by exposing clusters and cost

The big takeaways from the eval are that brokered video delivery in today's world is pretty broken and that a marketplace design would fix this by exposing clusters and cost.

Conclusion

- Identify challenges that brokers and CDNs create for each other from their lack of an interface
  - Requirements: traffic predictability, proper cluster pricing, and cluster-level optimization
- Examine the design space of CDN-broker interfaces
- Evaluate the efficacy of different designs
  - Marketplace design (“VDX”) is promising

In conclusion, in this work, we identify challenges that brokers and CDNs create for each other due to their lack of an interface, leading to three key requirements: traffic predictability, proper cluster pricing, and cluster-level optimization. We examine the design space of CDN-broker interfaces, and then evaluate them, finding that a marketplace design is promising.

Backup Slides…

Potential Problems

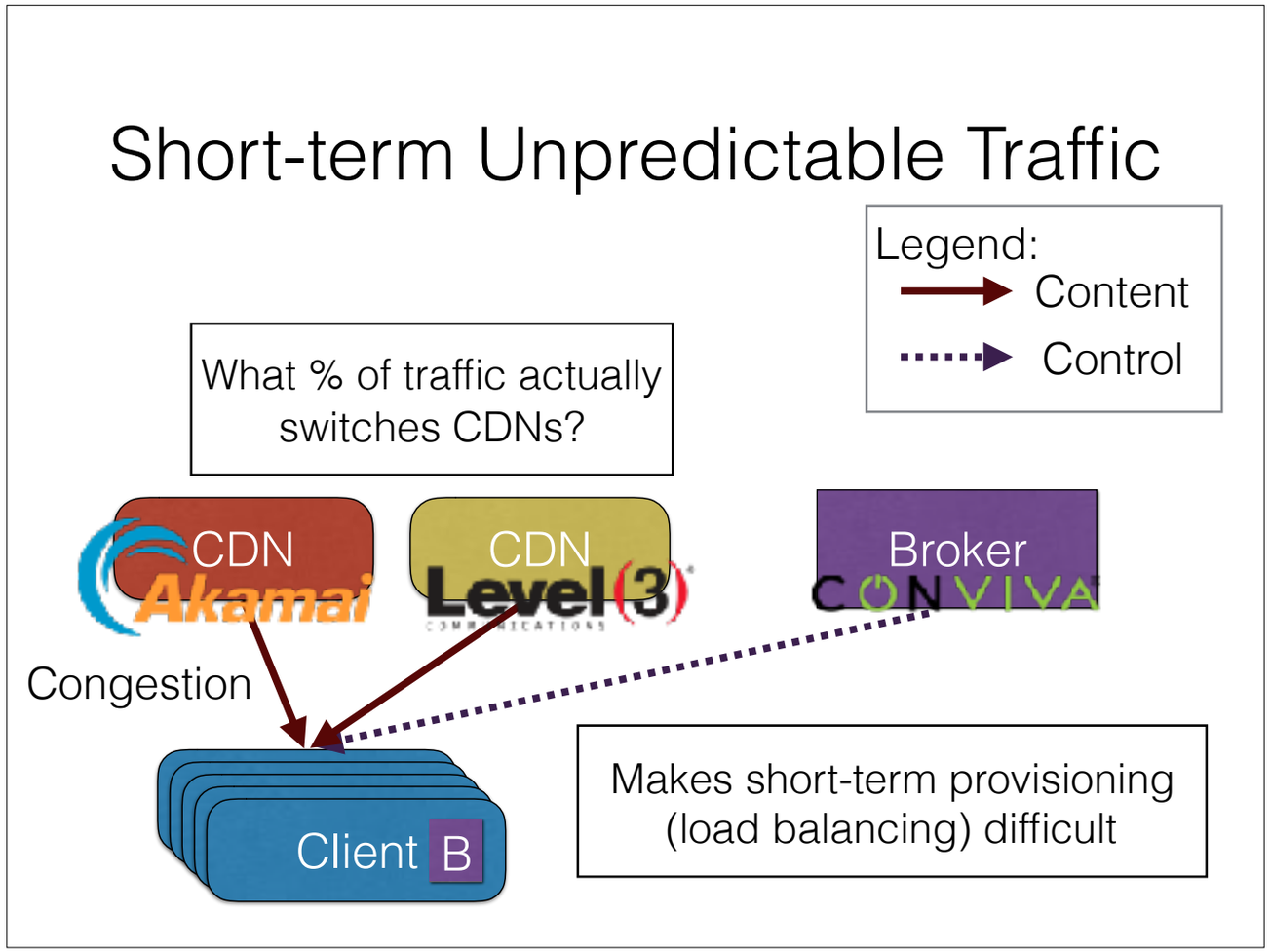
- Brokers cause CDN traffic to be unpredictable at **short** and long timescales making provisioning difficult
- When traffic is unpredictable, flat pricing model makes profits unpredictable

Broker

- Coarse CDN-level selection complicate meeting CP goals
- Incomplete measurements complicate meeting CP goals
- Difficult debugging

See Paper

Specifically, let's first focus on short term unpredictability, then talk about long term unpredictability.



Let’s look at this example. Here we see a client getting content from Akamai, but ** now there’s congestion. A broker can jump in (mid-session) ** and move this client ** to another CDN. Now imagine, instead of a single client ** this happens to a large number of clients. Clearly, moving large numbers of clients from one CDN to another ** makes short-term provisioning (i.e., load balancing) difficult for both CDNs. Does this problem actually happen in the wild though? Let’s look at data from a broker to find out ** what % of traffic actually switches CDNs.

Short-term Unpredictable Traffic

40%

of video delivery sessions switched CDNs during lifetime

Makes short-term provisioning (load balancing) difficult

We got data from a large broker involved in video delivery. The data contains video sessions from clients over one hour. We find that ** 40% of sessions switched CDNs during their lifetime. There’s a nice graph of this in the paper in detail. Thus when a broker is involved, ** CDN load balancing is potentially more difficult.

Potential Problems

- Brokers cause CDN traffic to be unpredictable at short and **long** timescales making provisioning difficult
- When traffic is unpredictable, flat pricing model makes profits unpredictable

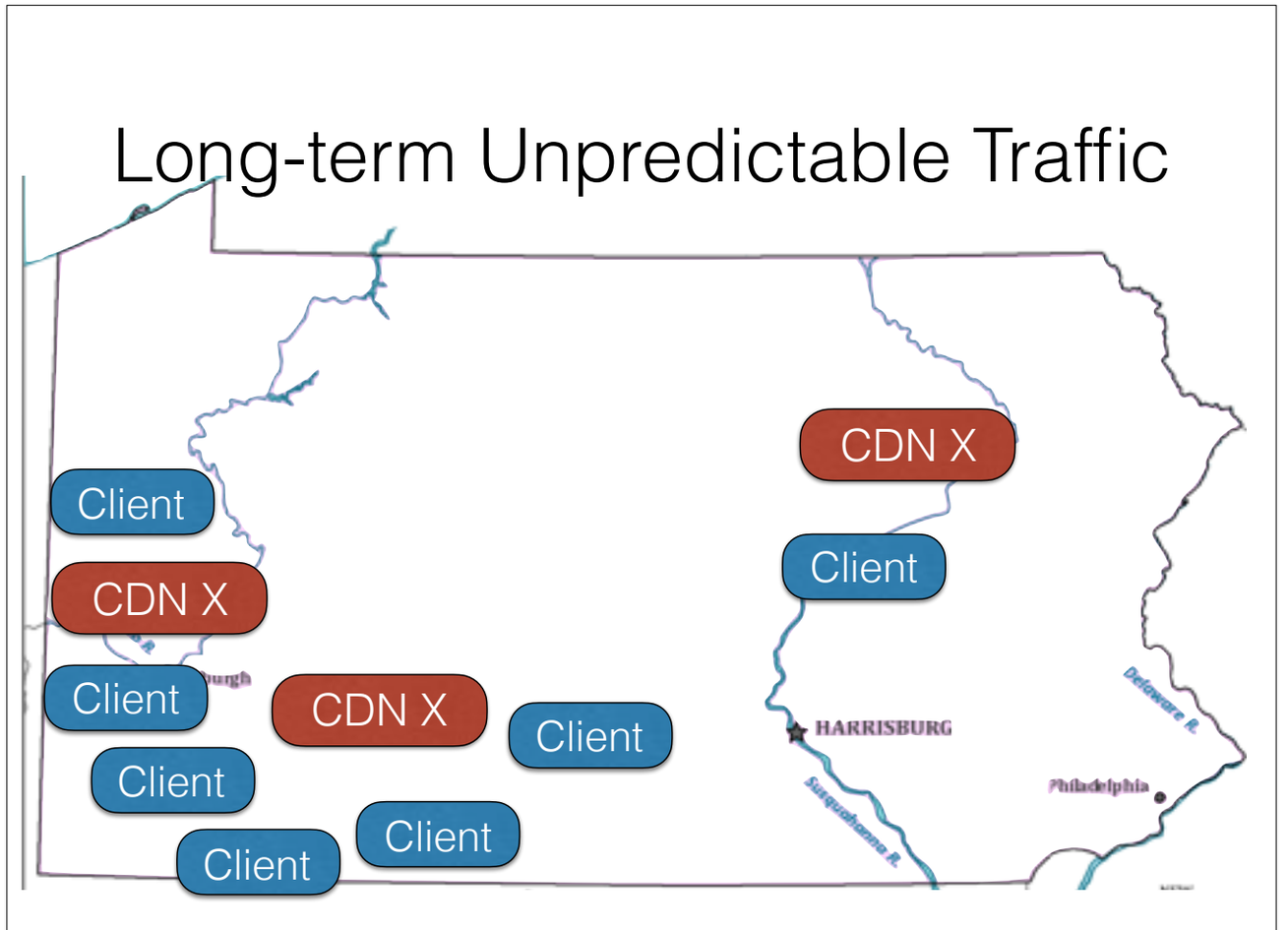
Broker

- Coarse CDN-level selection complicate meeting CP goals
- Incomplete measurements complicate meeting CP goals
- Difficult debugging

See Paper

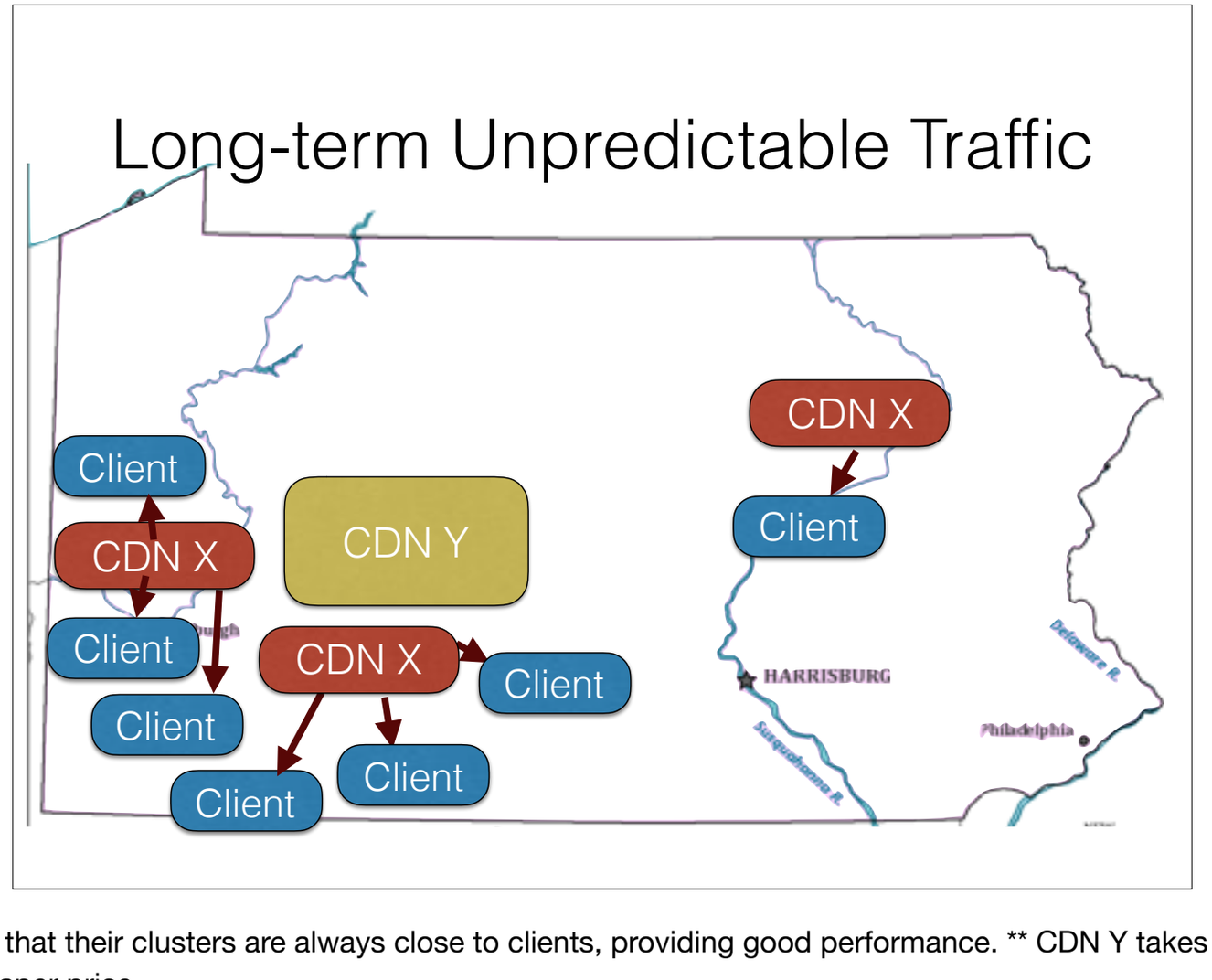
… let's look at how long term unpredictability.

Long-term Unpredictable Traffic

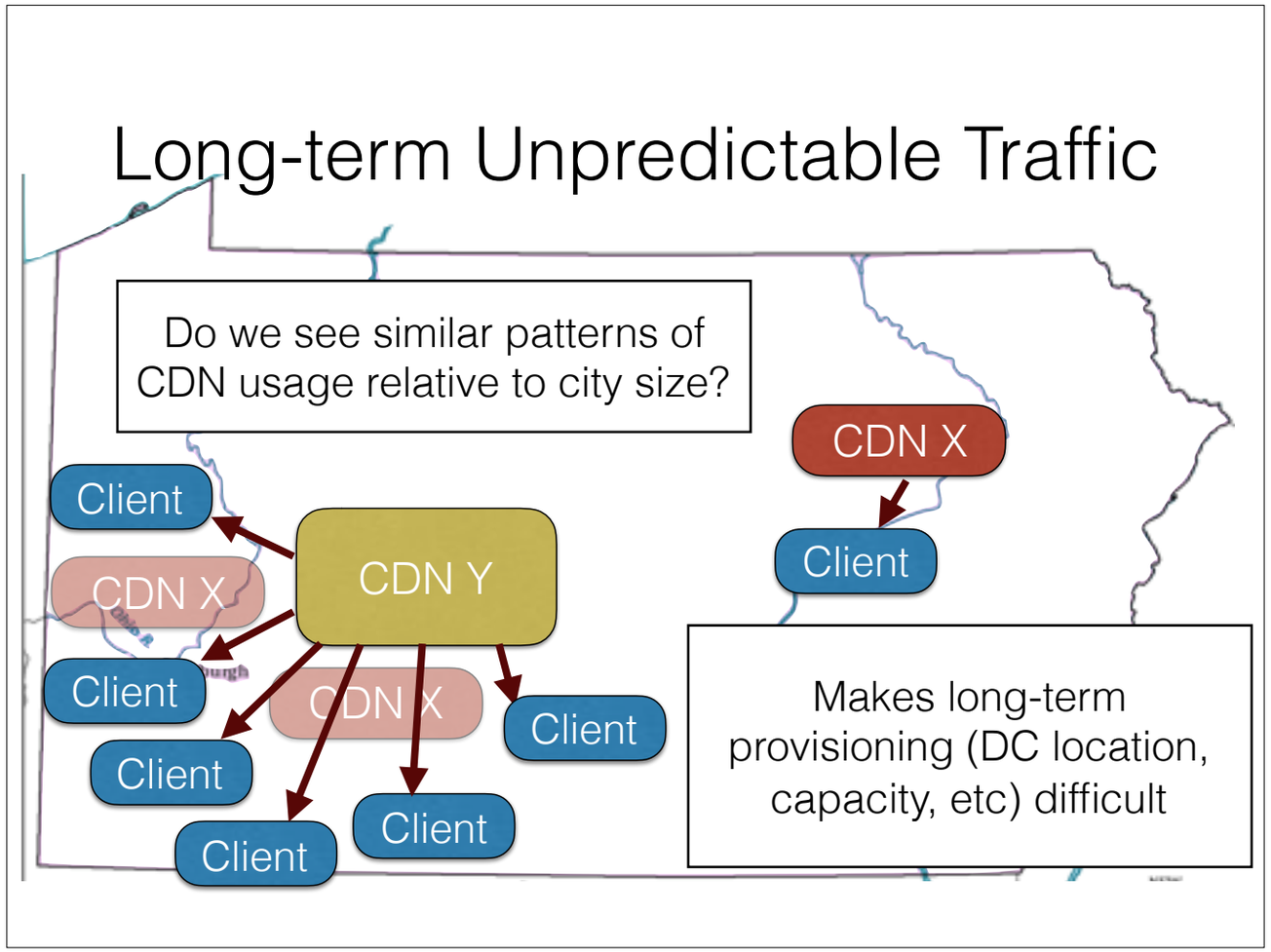


Let’s step through another hypothetical example. ** Here we see many clients in Pittsburgh, and ** one client in this rural area. ** ** Here we see CDN X’s clusters.

Long-term Unpredictable Traffic

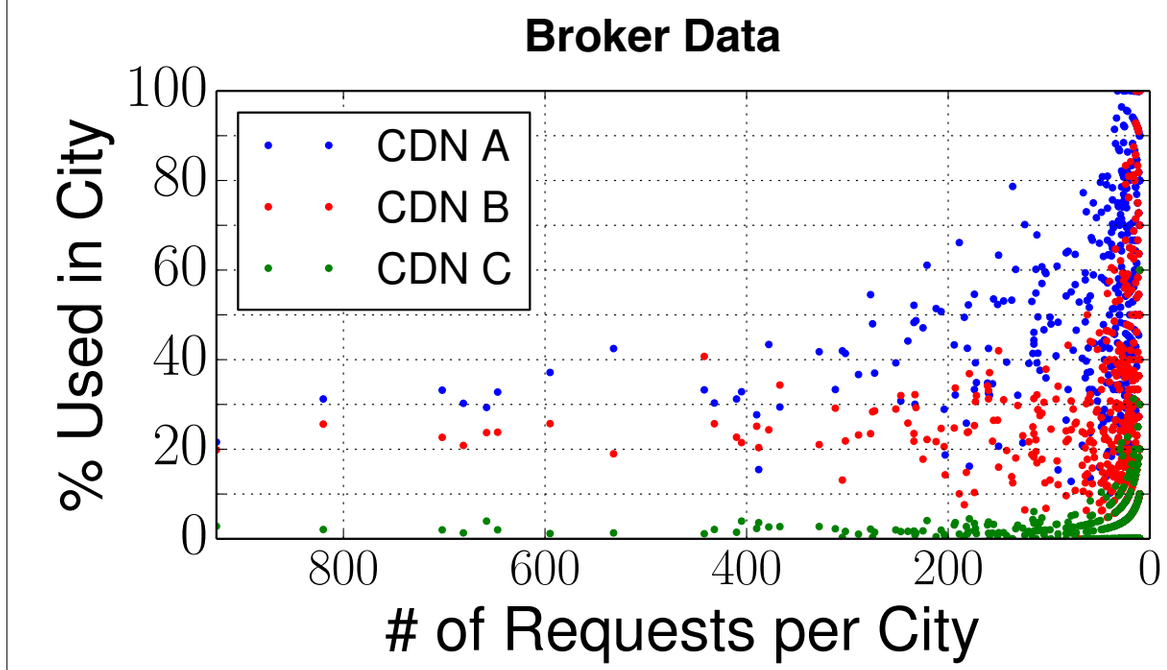


CDN X builds many delivery clusters so that their clusters are always close to clients, providing good performance. ** CDN Y takes an alternate approach, opting for fewer, high-capacity clusters with a cheaper price.



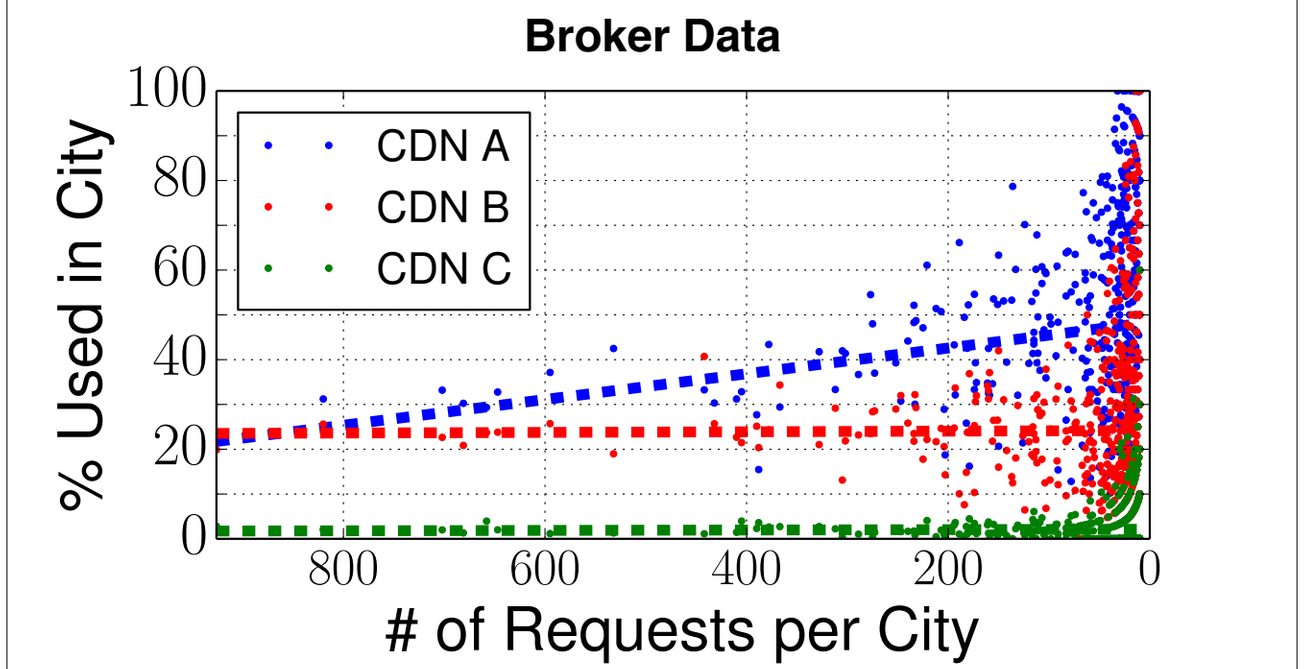
A broker sees that CDN Y can provide adequate performance at lower price, moving all the clients in the Pittsburgh area to CDN Y's cluster. In effect, the broker pushed CDN X out of the major city, only using it in rural areas. This goes against traditional provisioning wisdom— there is no longer positive correlation between number of clients in a region and the number of delivery clusters that should be placed in that region, ** in effect making long-term provisioning difficult (e.g., datacenter location, capacity planning, etc.). To see if this is an issue in practice, let's look at broker data ** to look for similar patterns in CDN usage relative to city size.

Long-term Unpredictable Traffic



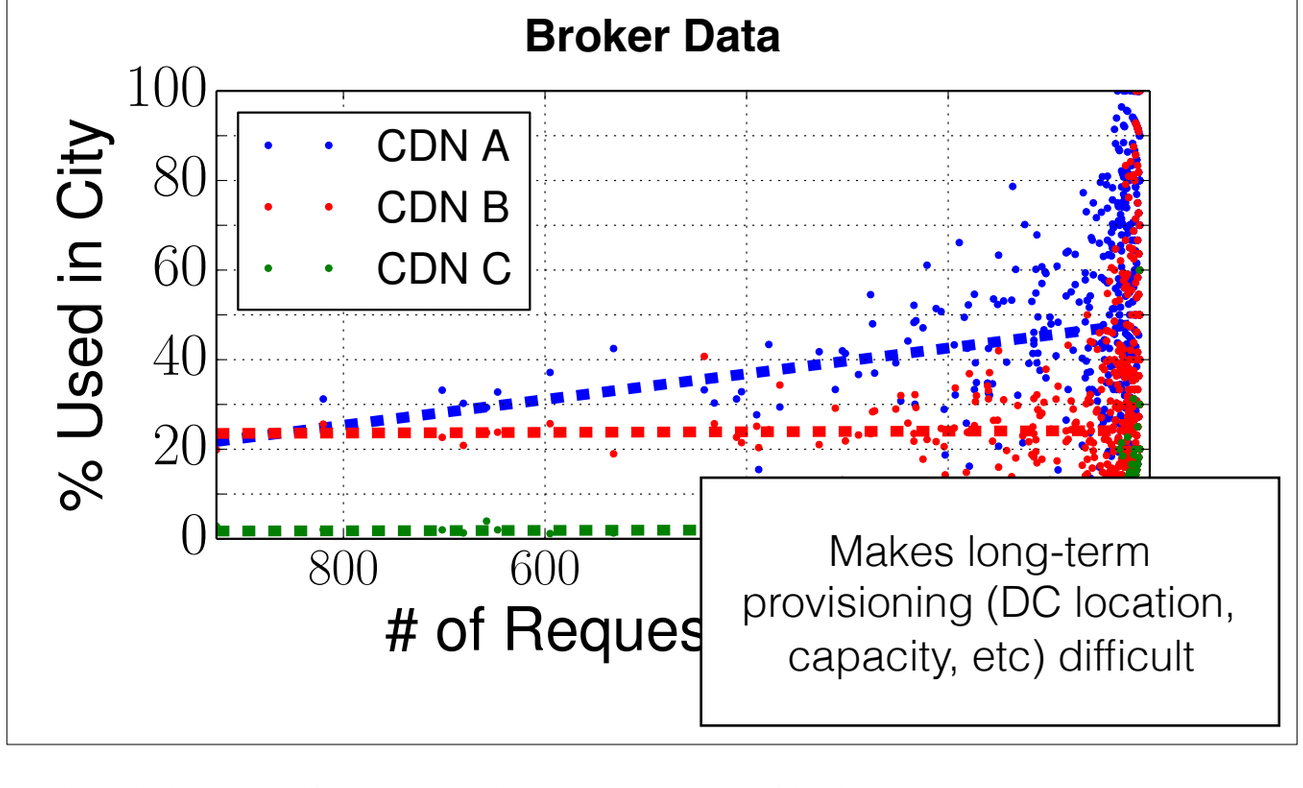
On the x-axis, we see cities sorted from large on the left to small on the right. On the y-axis we show which CDNs served clients in those cities as a percentage. The color series show the three CDNs explicitly labeled in our data as A, B, and C. The rest of the clients were served by “Other CDNs” which were grouped together in the data and are not plotted. To better understand the trends…

Long-term Unpredictable Traffic



… we plot best-fit lines over the data. The key takeaway is that CDN A is being used pushed towards specialized “small city” delivery. This CDN is similar to “CDN X” in our previous example: this CDN builds many delivery clusters both in large and small cities, but is more expensive when compared to its competitors. Thus, when other CDNs can provide adequate performance (in big cities— on the left), the more expensive CDN A is avoided. But in smaller cities (on the right), the performance gain of having a cluster closer outweighs the increase in price, thus CDN A is used more.

Long-term Unpredictable Traffic



Makes long-term provisioning (DC location, capacity, etc) difficult

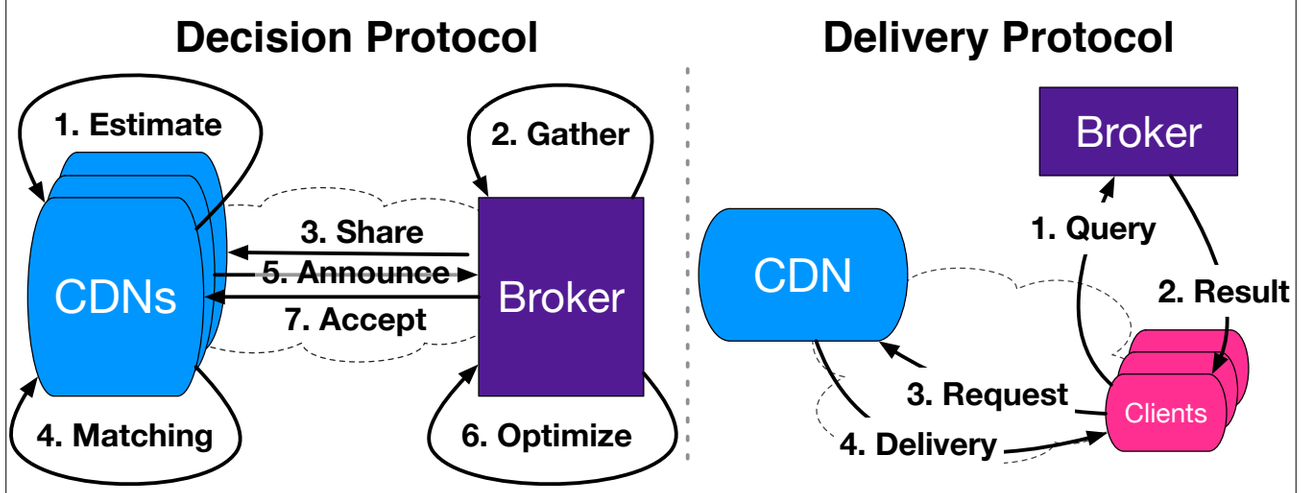
As mentioned, this make long-term provisioning difficult, as client location is no longer a good indicator for proper datacenter placement.

Design Space

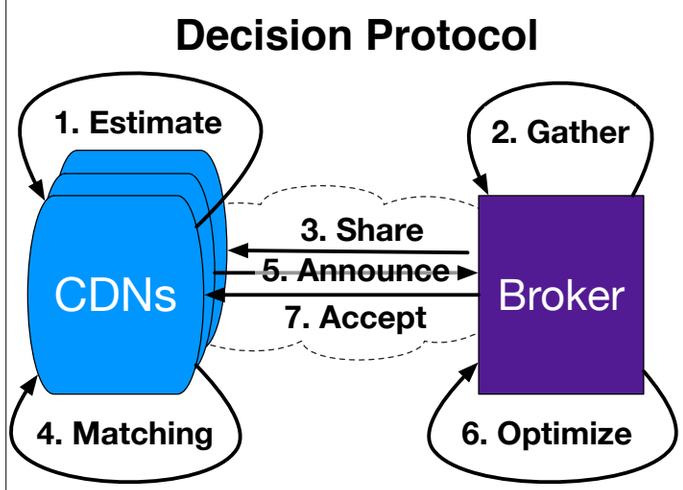
Decision Protocol

Delivery Protocol

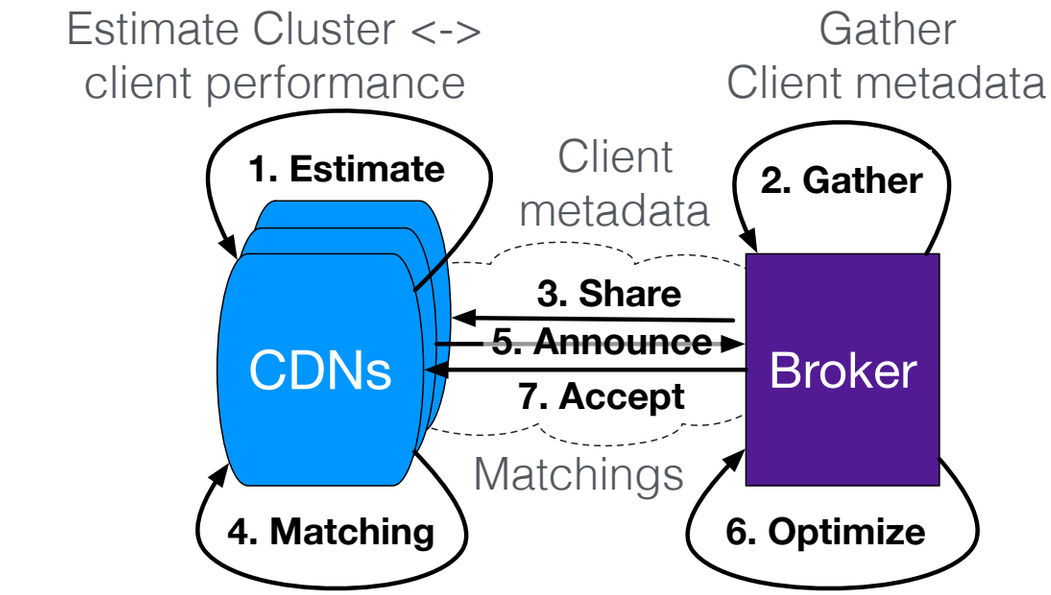
Design Space



Design Space

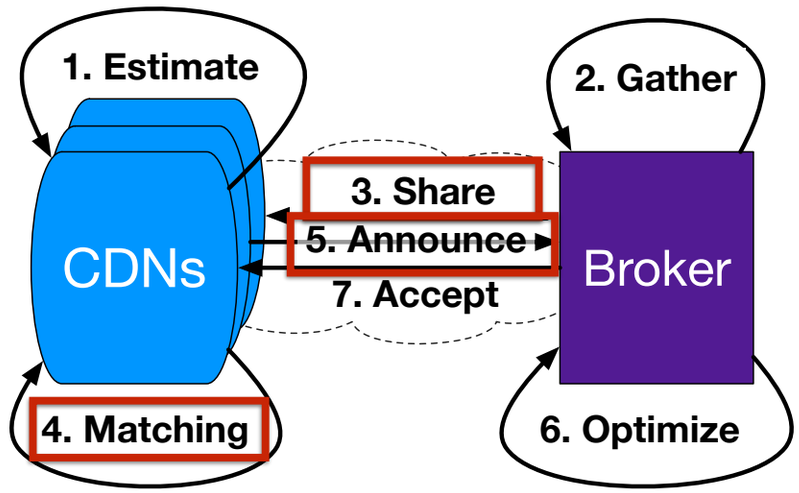


Decision Protocol



- Can we show all this more concretely? Little table with what data for example

Decision Protocol



Design Space

			Requirements		
Share	Matching	Announce	Cluster Optim.	Flexible Pricing	Traffic Predict.

- Can we describe all of this with pictures?

Design Space

	Share	Matching	Announce	Requirements: Cluster Optim.	Flexible Pricing	Traffic Predict.
Today		Single-Cluster		-	-	-

Design Space

	Share	Matching	Announce	Requirements: Cluster Optim.	Flexible Pricing	Traffic Predict.
Today		Single-Cluster		-	-	-
Multicluster		Multi-Cluster	Perf.	+	-	-

Brokers can do finer-grain optimization

Design Space

	Share	Matching	Announce	Requirements: Cluster Optim.	Flexible Pricing	Traffic Predict.
Today		Single-Cluster		-	-	-
Multicluster		Multi-Cluster	Perf.	+	-	-
Dynamic Pricing		Single-Cluster	Cost	-	+	-

Fixes CDN cost issues

Design Space

	Share	Matching	Announce	Requirements: Cluster Optim.	Flexible Pricing	Traffic Predict.
Today		Single-Cluster		-	-	-
Multicluster		Multi-Cluster	Perf.	+	-	-
Dynamic Pricing		Single-Cluster	Cost	-	+	-
Dynamic-Multicluster		Multi-Cluster	Cost, Perf.	+	+	-

Fixes both, but no traffic predictability

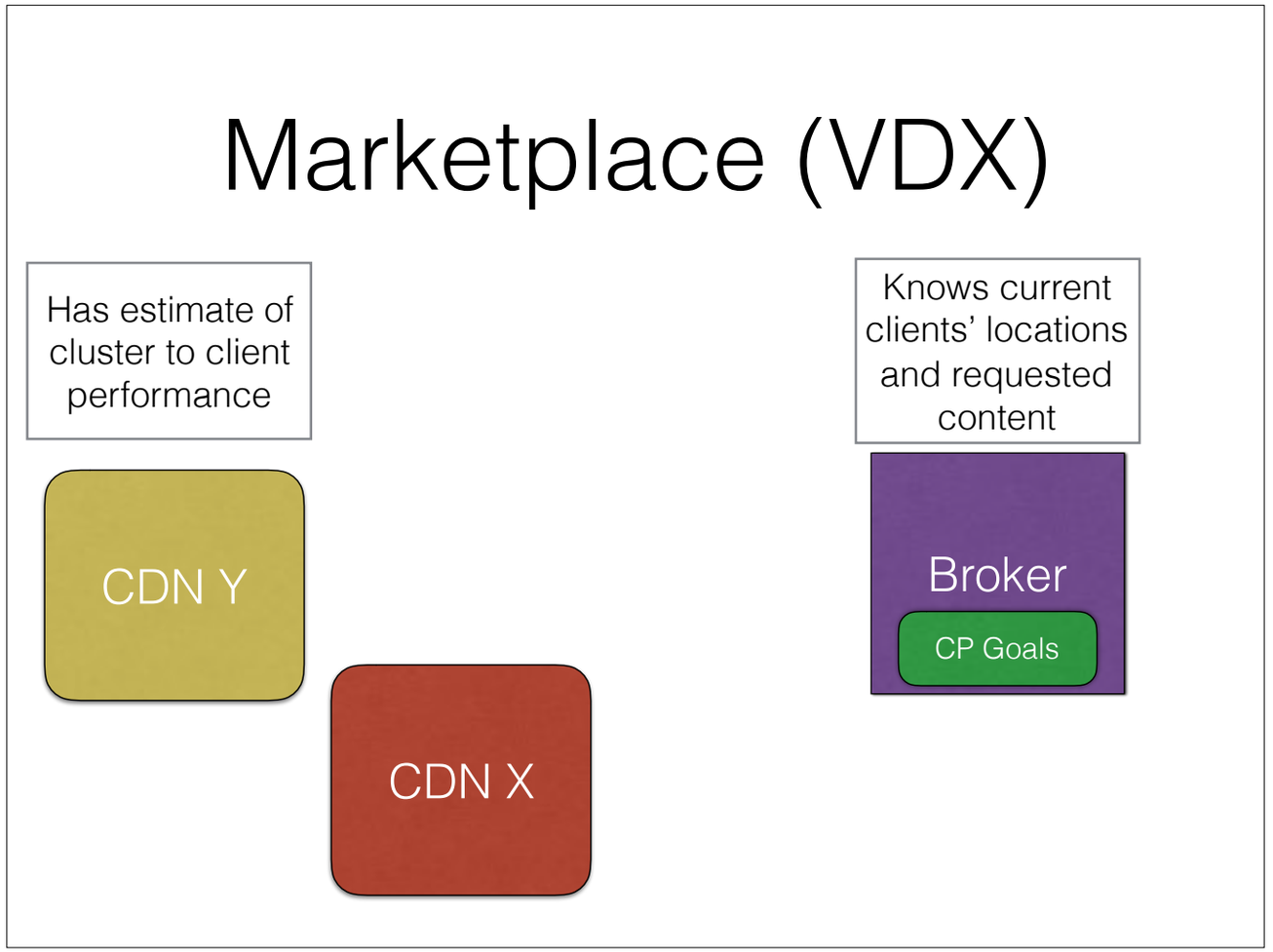
Design Space

				Requirements		
	Share	Matching	Announce	Cluster Optim.	Flexible Pricing	Traffic Predict.
Today		Single-Cluster		-	-	-
Multicluster		Multi-Cluster	Perf.	+	-	-
Dynamic Pricing		Single-Cluster	Cost	-	+	-
Dynamic-Multicluster		Multi-Cluster	Cost, Perf.	+	+	-
Marketplace	Clients	Multi-Cluster	Cost, Perf., Capacities	+	+	weak

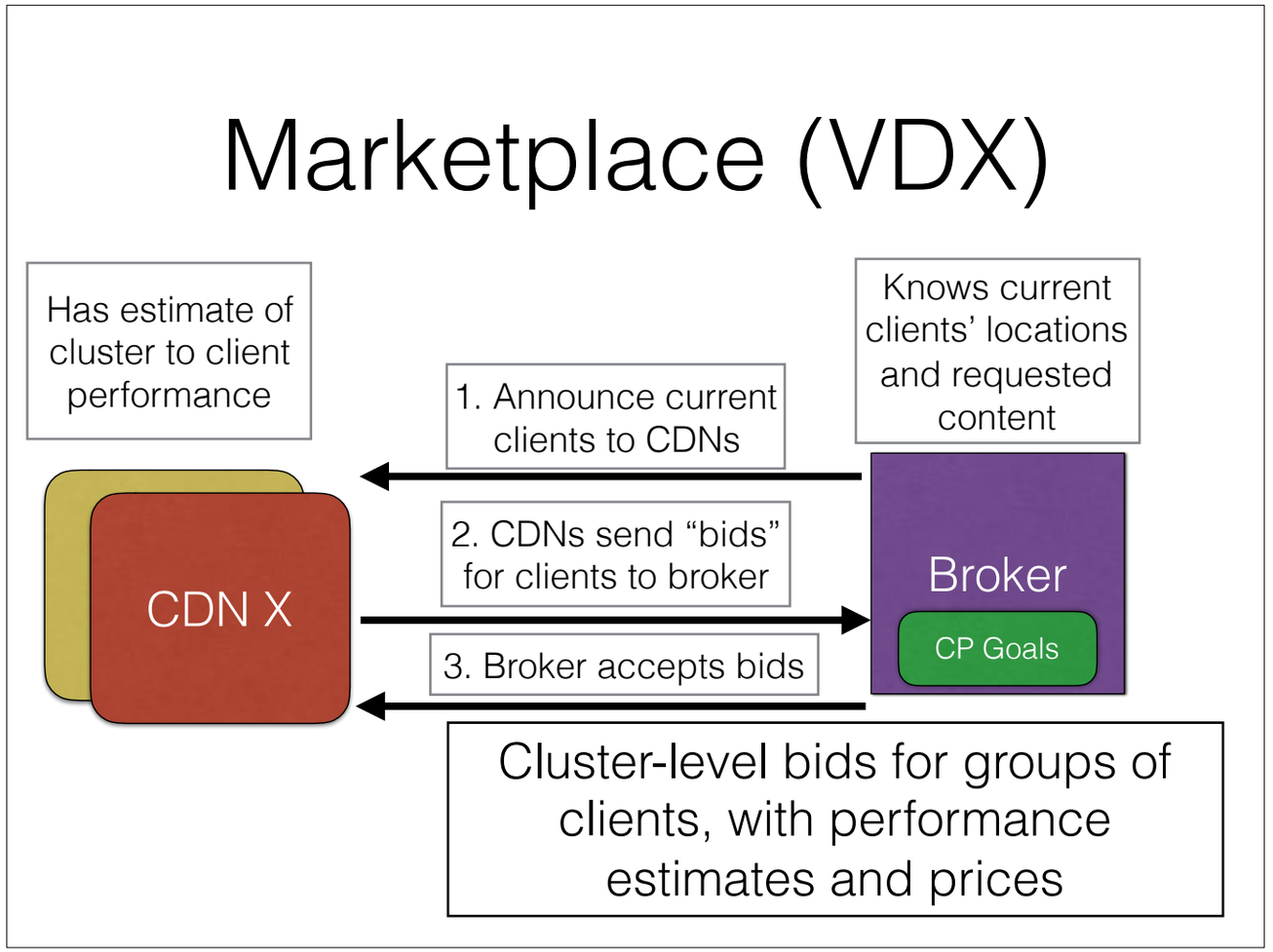
Design Space

				Requirements		
	Share	Matching	Announce	Cluster Optim.	Flexible Pricing	Traffic Predict.
Today		Single-Cluster		-	-	-
Multicluster		Multi-Cluster	Perf.	+	-	-
Dynamic Pricing		Single-Cluster	Cost	-	+	-
Dynamic-Multicluster		Multi-Cluster	Cost, Perf.	+	+	-
Marketplace	Clients	Multi-Cluster	Cost, Perf., Capacities	+	+	weak

- Leave it a little more open; which one does best?

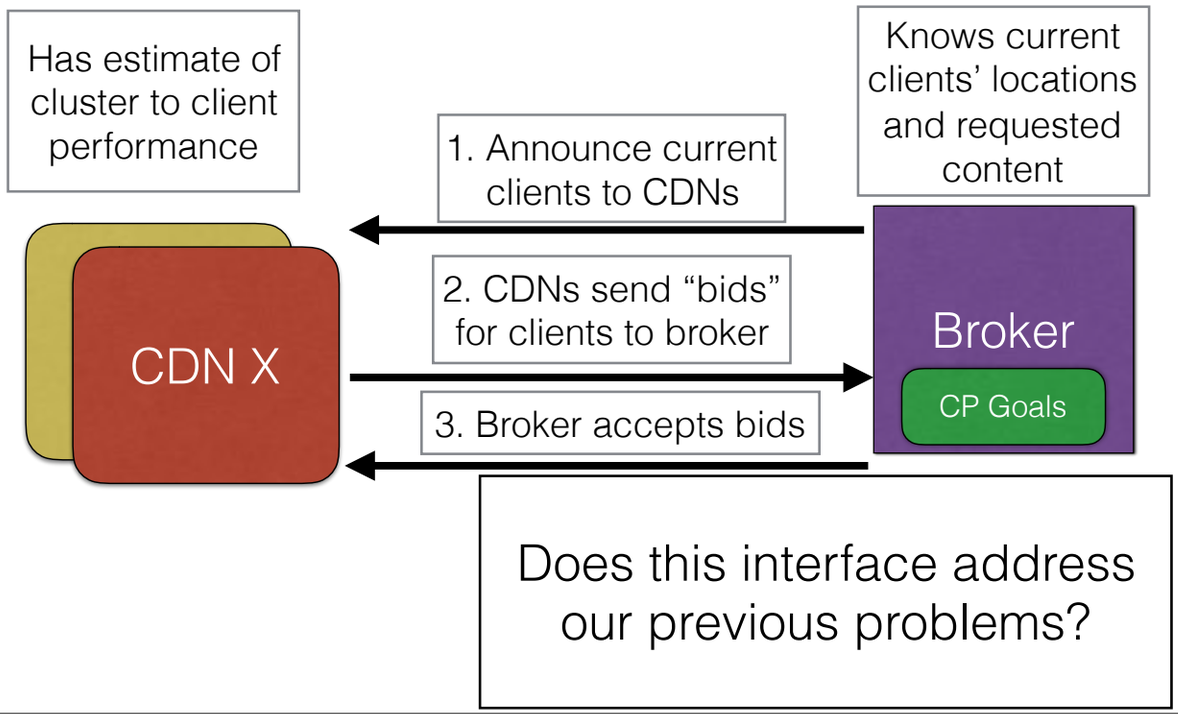


… an ad exchange. As with before, ** CDNs still estimate cluster to client performance. Brokers still ** know about current clients' locations and what content they've requested. But, from here things differ.



Our CDN-broker interface is a control plane protocol that runs in the background periodically. Conceptually, it consists of three stages, here drawn as arrows. First, ** the broker announces the current set of clients to all CDNs. Second, ** the CDNs send "bids" for clients to the broker. These bids ** are done per cluster for groups of clients, with performance estimates and some notion of price. Finally, ** the broker sends back a list of accepted bids to the CDNs.

Marketplace (VDX)



Let's look at how this new proposal addresses the problems we saw previously.

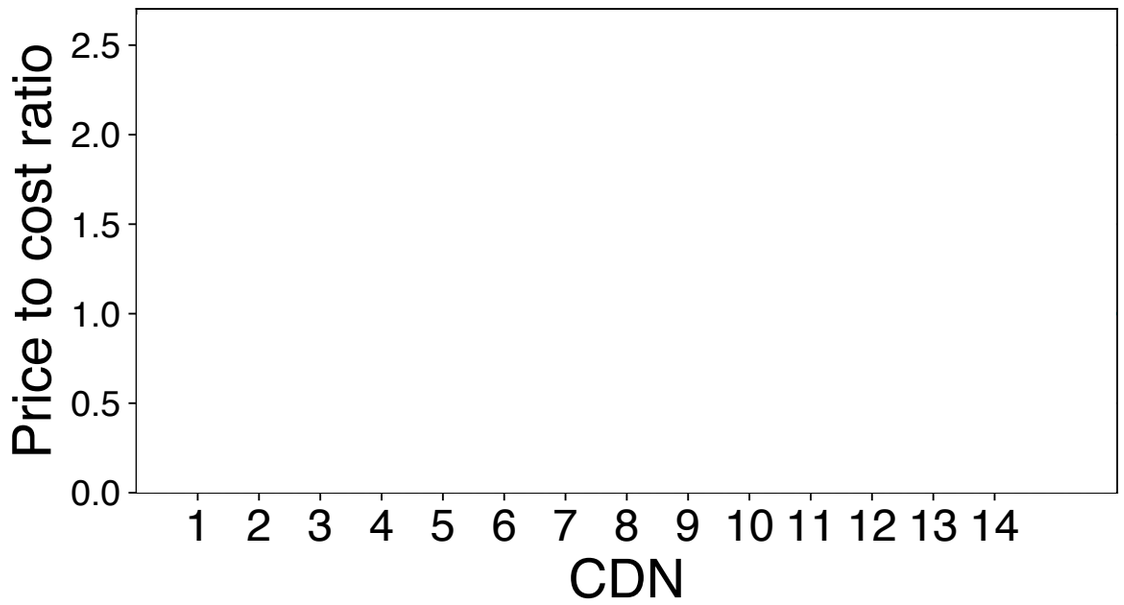
Simulation Model

- **Clients' location/bitrate** from broker data
- **CDN clusters** from our CDN data + PeeringDB
- CDN/client location **perform. estimates** from CDN
- CDN locations have **bw cost + colo cost**
  - bw cost chosen from CDN data
  - colo cost is similar but decreases when more CDNs use that location

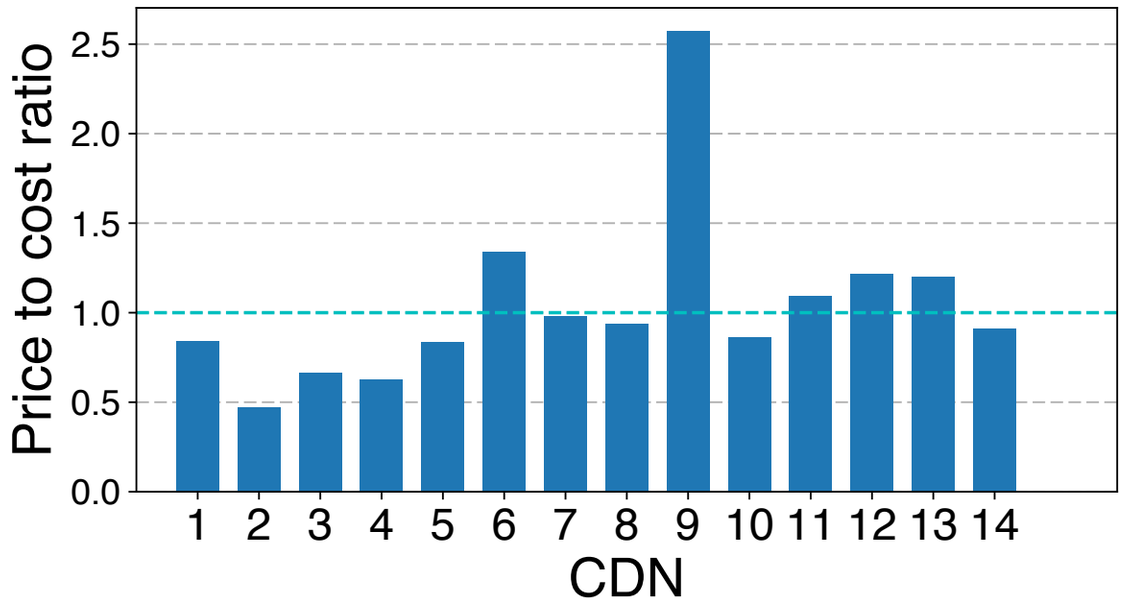
Simulation Model

- **Contract prices** (for strawmen) estimated from average price-per-bit if CDN were offered all clients
- **CDN capacity** assigned based on load if offered all clients
- **Broker optimize client matching** with simple ILP
- CDNs select **candidate clusters** with performance estimate <2x worse than best cluster.
- CDN offered **bids sorted by cost**

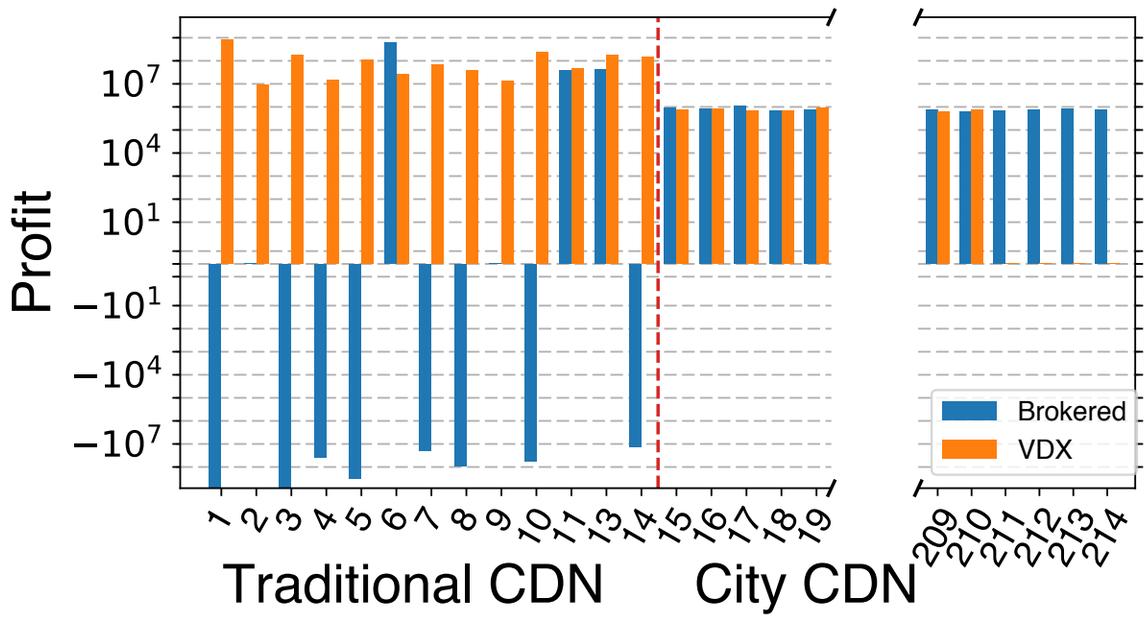
Data Driven:
CDN Comparisons Today



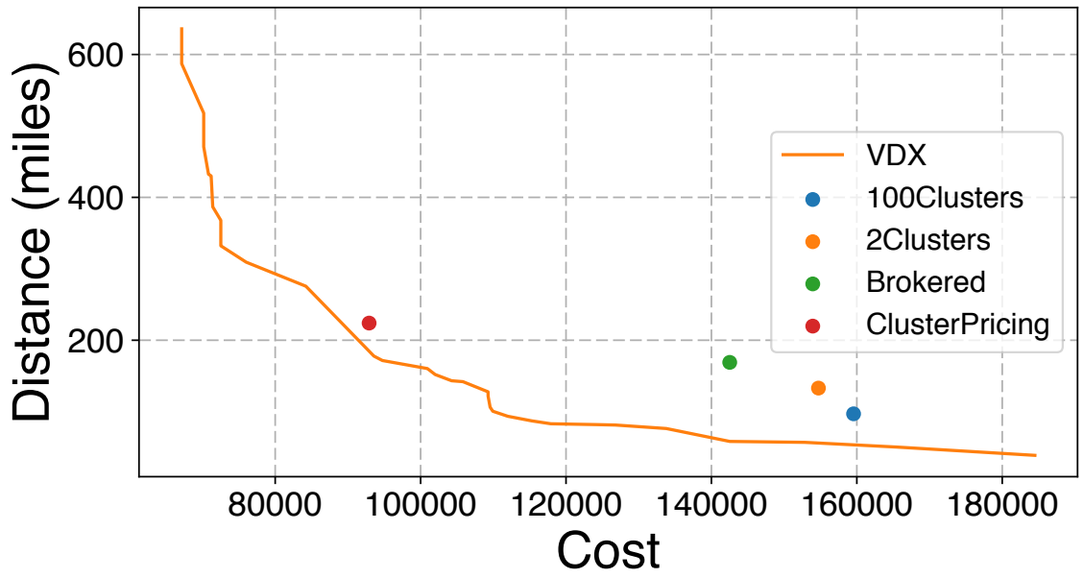
Data Driven:
CDN Comparisons Today



Scenarios:
City-Centric CDNs



Microbenchmarks: Performance / Cost Tradeoff



Questions from Audience

- Why would this work when CDN federation has not?
  - Broker & CDNs both serve CP. It's a tweak to an existing market, not creation of a new one.
- Does the bidding protocol make actual content delivery slower?
  - No, the bidding protocol is a periodic offline control plane protocol. The data plane is still the same as today.
- Why "auctions"? Why not dynamic pricing?
  - Dynamic pricing can be just as difficult (e.g., stability, convergence, fairness), rearchitecting might be the best approach. Also, there are other gains we didn't touch on in the talk (e.g., ability to use CDN clusters that the broker can't currently see— see paper)