Comparing PubSubHubbub and Twitter

Introduction to Information Retrieval
INF 141/ CS 121
Donald J. Patterson

Remember this?

Politeness when crawling

Statistics for:
djp3.net

Summary
When:
Monthly history
Days of month
Days of week
Hours
Who:
Countries
Full list
Hosts
Full list
Last visit
Unresolved IP Address
Robots/Spiders visitors
Full list
Last visit
Navigation:
Visits duration
File type
Viewed
Full list
Entry
Exit
Operating Systems
Versions
Unknown
Browsers
Versions
Unknown
Referers:
Origin
Refering search engines
Refering sites
Search
Search Keyphrases
Search Keywords

Last Update: 14 Jan 2008 - 02:59
Reported period: - Year - 2007 OK

speakeasy

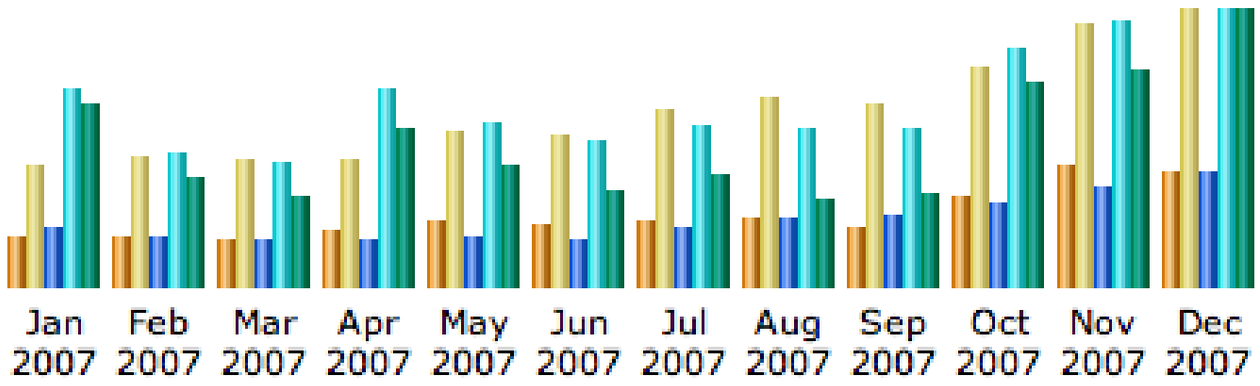
Back to main page

Robots/Spiders visitors

30 different robots	Hits	Bandwidth	Last visit
Googlebot	1393868+104	5.11 GB	31 Dec 2007 - 23:50
Inktomi Slurp	36668+221	554.25 MB	31 Dec 2007 - 23:55
MSNBot	19522+2	699.90 MB	28 Dec 2007 - 08:01
Unknown robot (identified by 'crawl')	15949+13	89.34 MB	31 Dec 2007 - 22:24
AskJeeves	7016+1	106.29 MB	31 Dec 2007 - 23:49
Google AdSense	2701	100.26 MB	31 Dec 2007 - 22:10
psbot	2268+1	80.48 MB	31 Dec 2007 - 09:59
Unknown robot (identified by 'robot')	930+1	19.10 MB	31 Dec 2007 - 09:34
Turn It In	350+1	6.32 MB	03 Sep 2007 - 15:44
BaiDuSpider	300	10.22 MB	26 Nov 2007 - 07:32
GigaBot	243	5.27 MB	30 Dec 2007 - 05:06
Scooter	90+3	288.75 KB	27 Nov 2007 - 14:30
PhpDig	91	2.28 MB	21 Oct 2007 - 09:51
WISENutbot	76	1.94 MB	13 Jan 2007 - 14:04
Magpie	25	43.48 KB	24 Dec 2007 - 00:51
Unknown robot (identified by hit on 'robots.txt')	0+16	4.38 KB	14 Nov 2007 - 03:43
EchO!	14	287.09 KB	27 Dec 2007 - 13:56
Internet Shinchakubin	13	385.03 KB	27 Nov 2007 - 15:23
BBot	10	146.35 KB	13 Jun 2007 - 15:17
arks	8	142.24 KB	27 Nov 2007 - 12:25
MSIECrawler	8	263.02 KB	26 Dec 2007 - 11:16

Politeness?

Monthly history

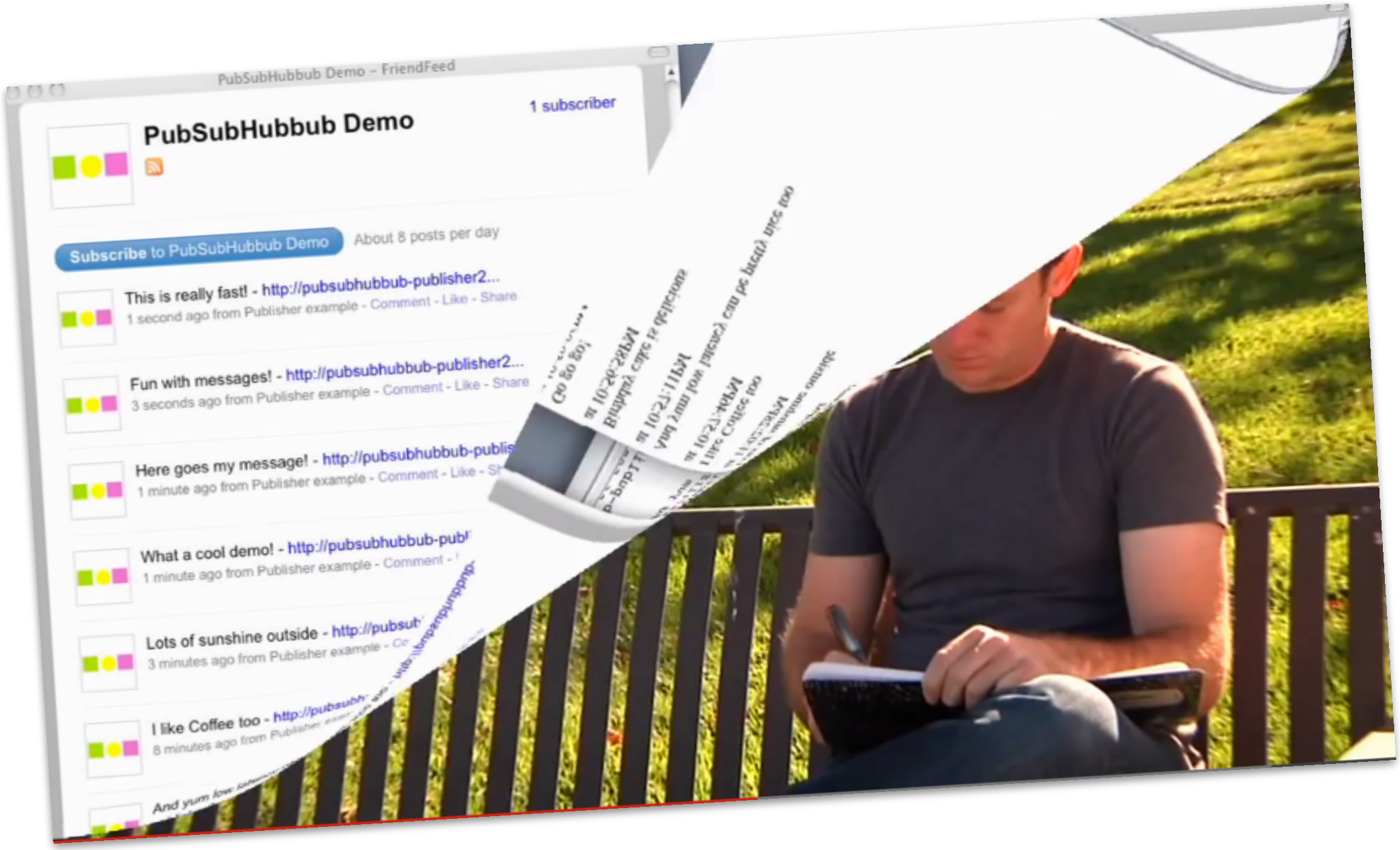


Month	Unique visitors	Number of visits	Pages	Hits	Bandwidth
Jan 2007	1221	2946	8938	30536	699.28 MB
Feb 2007	1179	3099	7852	20475	415.75 MB
Mar 2007	1120	3063	7099	18978	350.88 MB
Apr 2007	1362	3067	7175	30320	599.91 MB
May 2007	1612	3746	7584	25114	469.32 MB
Jun 2007	1474	3662	7138	22292	370.11 MB
Jul 2007	1592	4210	9165	24766	430.61 MB
Aug 2007	1658	4567	10600	24142	336.08 MB
Sep 2007	1458	4403	11149	24414	356.60 MB
Oct 2007	2148	5299	12877	36427	783.78 MB
Nov 2007	2890	6317	15300	40487	833.75 MB
Dec 2007	2748	6631	17553	42281	1.03 GB
Total	20462	51010	122430	340232	6.55 GB

The problem

- Crawling incurs a huge waste of resources:
  - server bandwidth
  - crawler bandwidth
  - server CPU
  - electricity
  - hard drive space (for logs)

The problem


PubSubHubbub Demo – FriendFeed
PubSubHubbub Demo
1 subscriber
Subscribe to PubSubHubbub Demo
About 8 posts per day
This is really fast! - http://pubsubhubbub-publisher2...
1 second ago from Publisher example - Comment - Like - Share
Fun with messages! - http://pubsubhubbub-publisher2...
3 seconds ago from Publisher example - Comment - Like - Share
Here goes my message! - http://pubsubhubbub-publis
1 minute ago from Publisher example - Comment - Like - Sh
What a cool demo! - http://pubsubhubbub-publ
1 minute ago from Publisher example - Comment -
Lots of sunshine outside - http://pubsub
3 minutes ago from Publisher example - Co
I like Coffee too - http://pubsubh
8 minutes ago from Publisher

PubSubHubbub

- Basic idea is to eliminate polling sites for changes
  - = eliminate crawling web sites blindly

PubSubHubbub

- A "topic" declares its PuSH server
  - A topic is a URL
    - Could be a webpage
    - Could be an {Atom or RSS} XML file associated with a blog
    - Could be anything that is referenced by a URL

- A PuSH server is a "hub"

PubSubHubbub

- A PuSH server is a "hub"
  - a 3rd party server
  - manages your subscribers
  - receives your updates
  - rebroadcasts your updates

PubSubHubbub

- A "topic" can declare who its PuSH server:
  - In the HTTP response
    - rel=hub (indicating URL of hub)
    - rel=self (indicating URL of self)
  - If it is an html page, in can be in the header

```
<link rel="hub" href="http://myhub.example.com/endpoint" />
```

```
<atom:link rel="hub" href="http://pubsubhubbub.appspot.com"/>
<atom:link rel="hub" href="http://superfeedr.com/hubbub"/>
```

PubSubHubbub

- A subscriber first gets the “topic” as always
  - for example a news feed reader
  - a webpage

PubSubHubbub

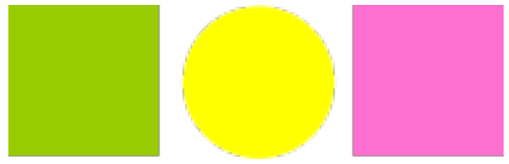
- A subscriber then subscribes to the "hub"
  - The hub then tells the subscriber when there is an update
  - This avoids endless polling the feed by the subscriber for changes

PubSubHubbub

- When there is an update, the publisher tells the hub

PubSubHubbub

- The hub gets the url and sends it to the subscribers

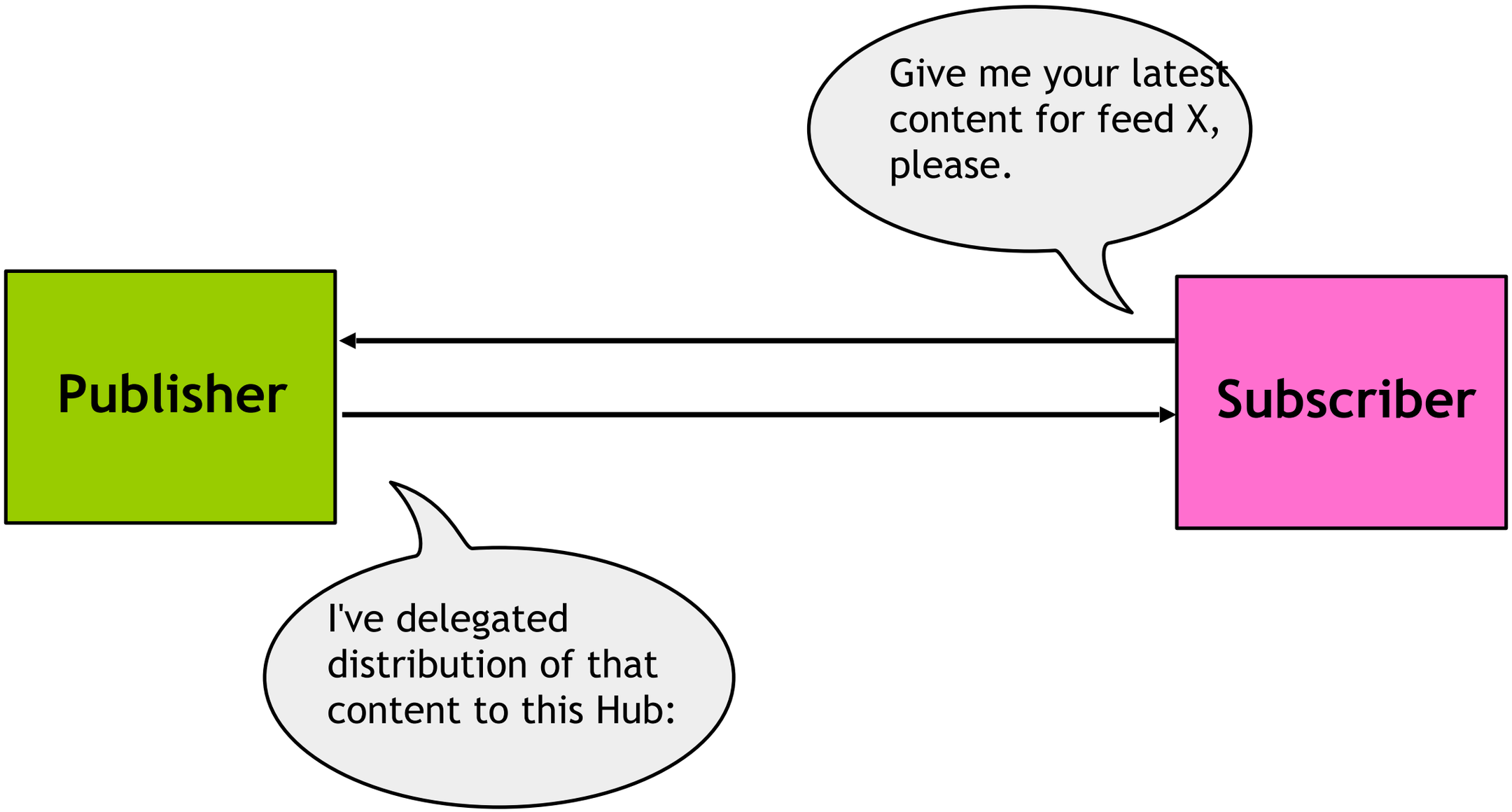


PubSubHubbub

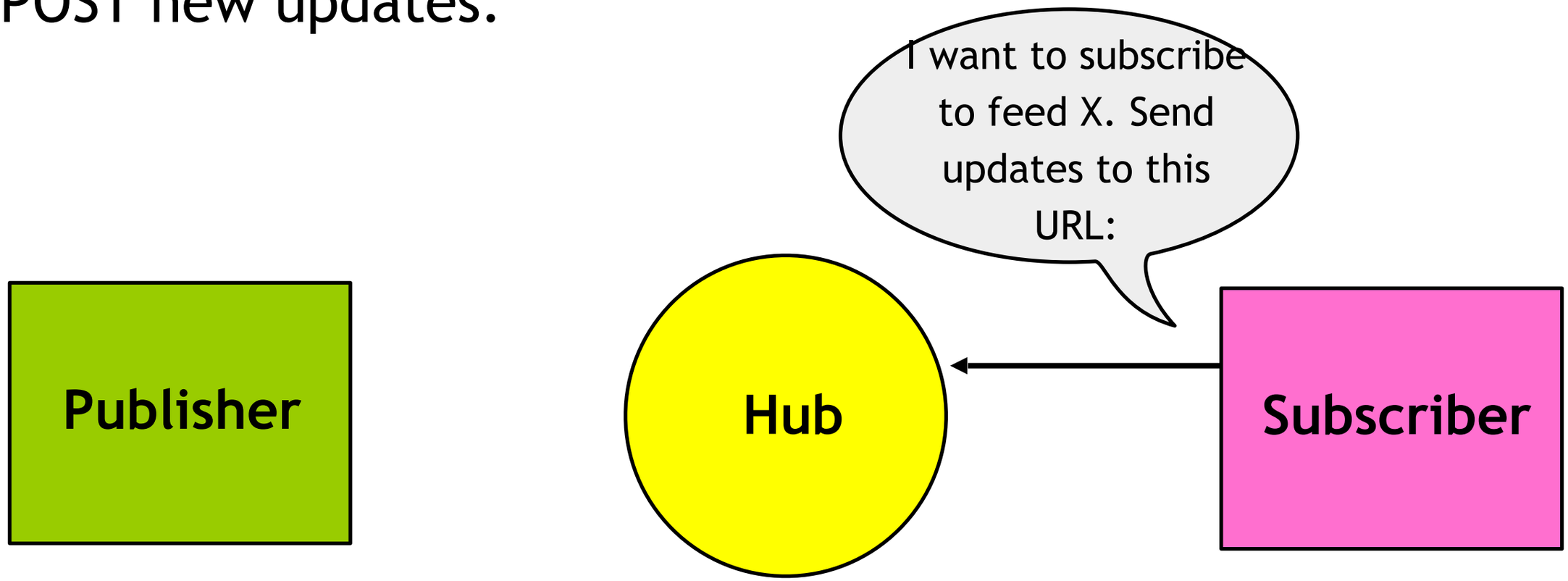
Subscription flow

Draft 0.1

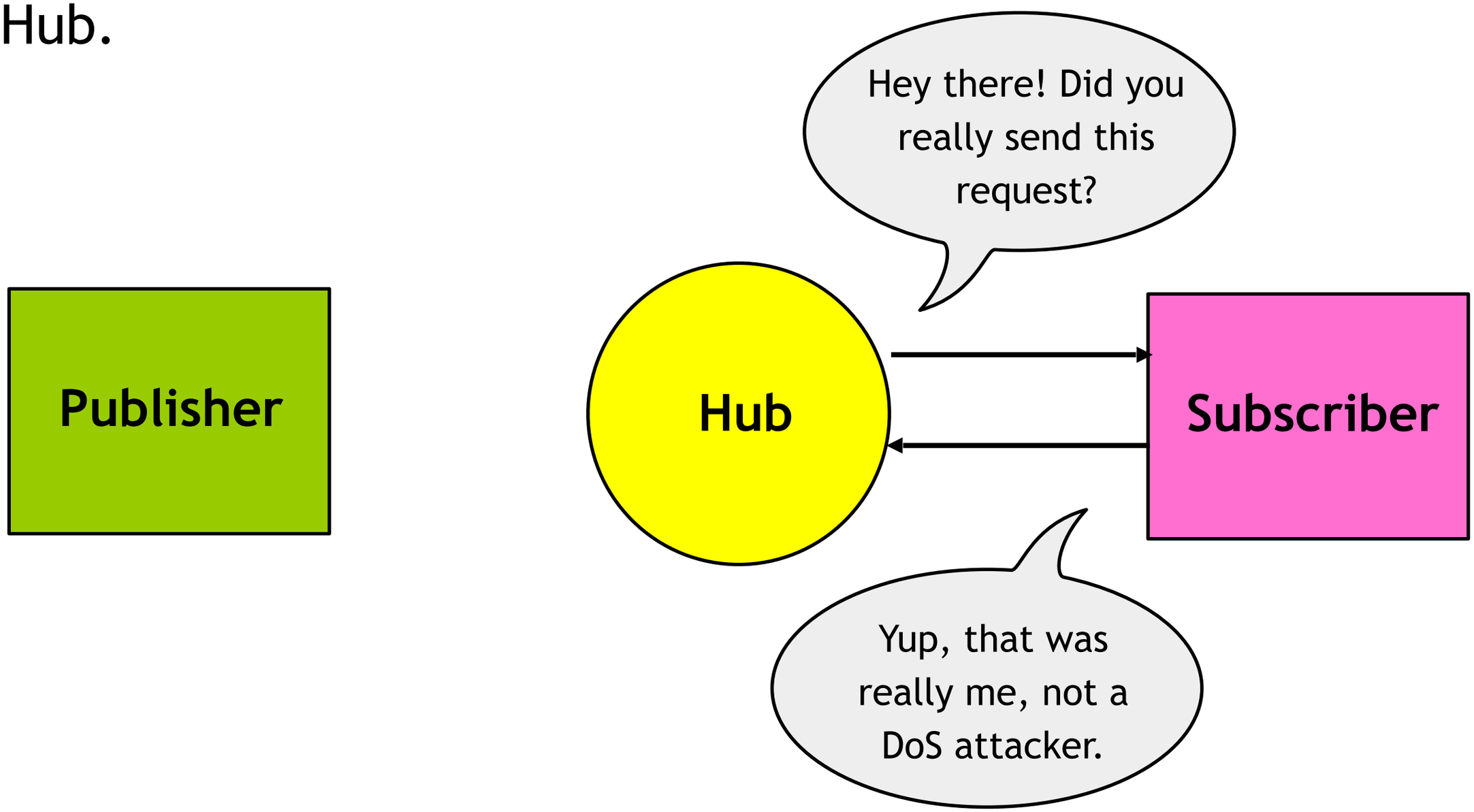
1. Subscriber polls Publisher's feed. The feed contains a forward link to the Hub.



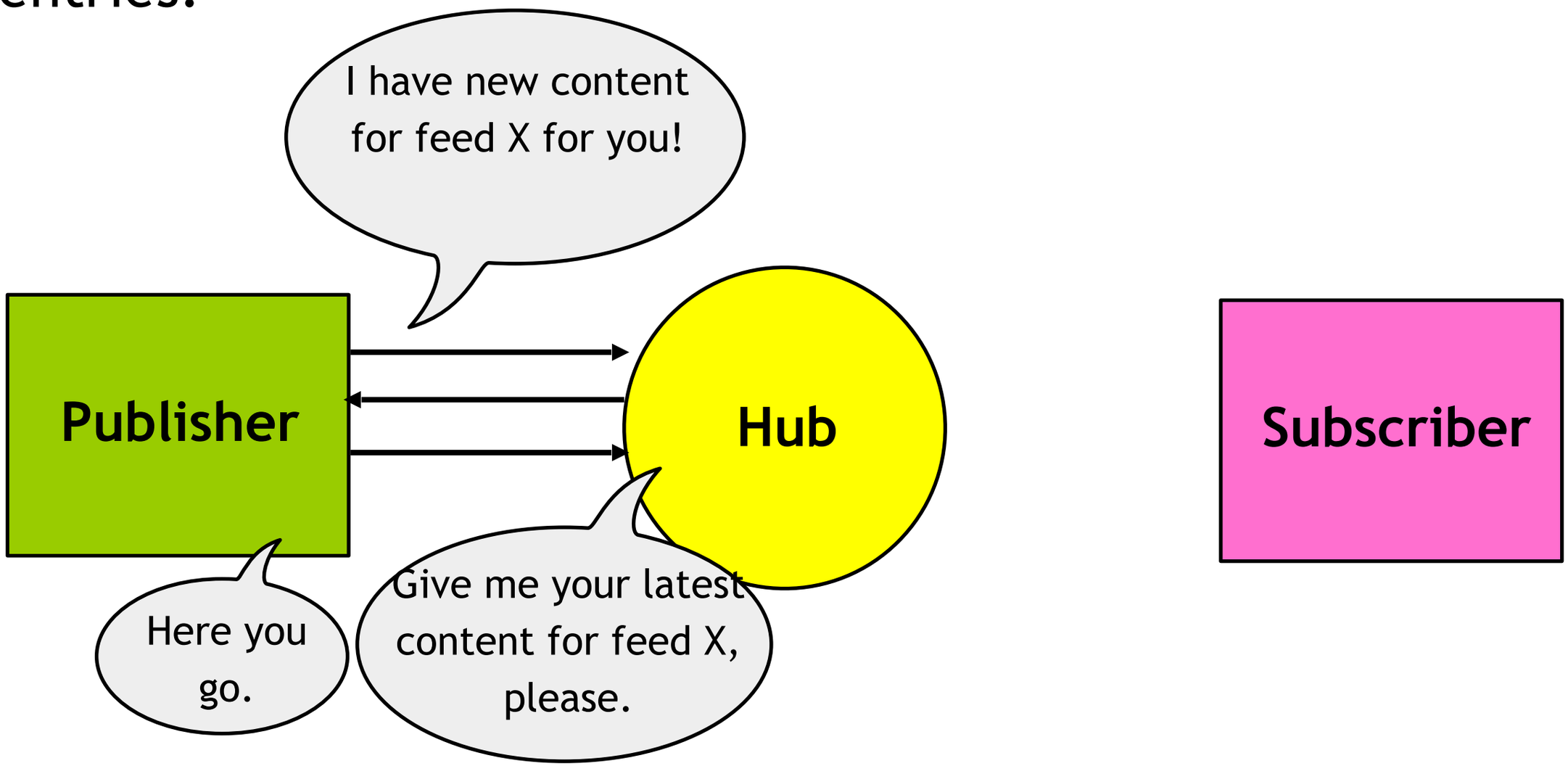
2. Subscriber POSTs subscription request to the Hub. The request contains the endpoint URL where the Hub should POST new updates.



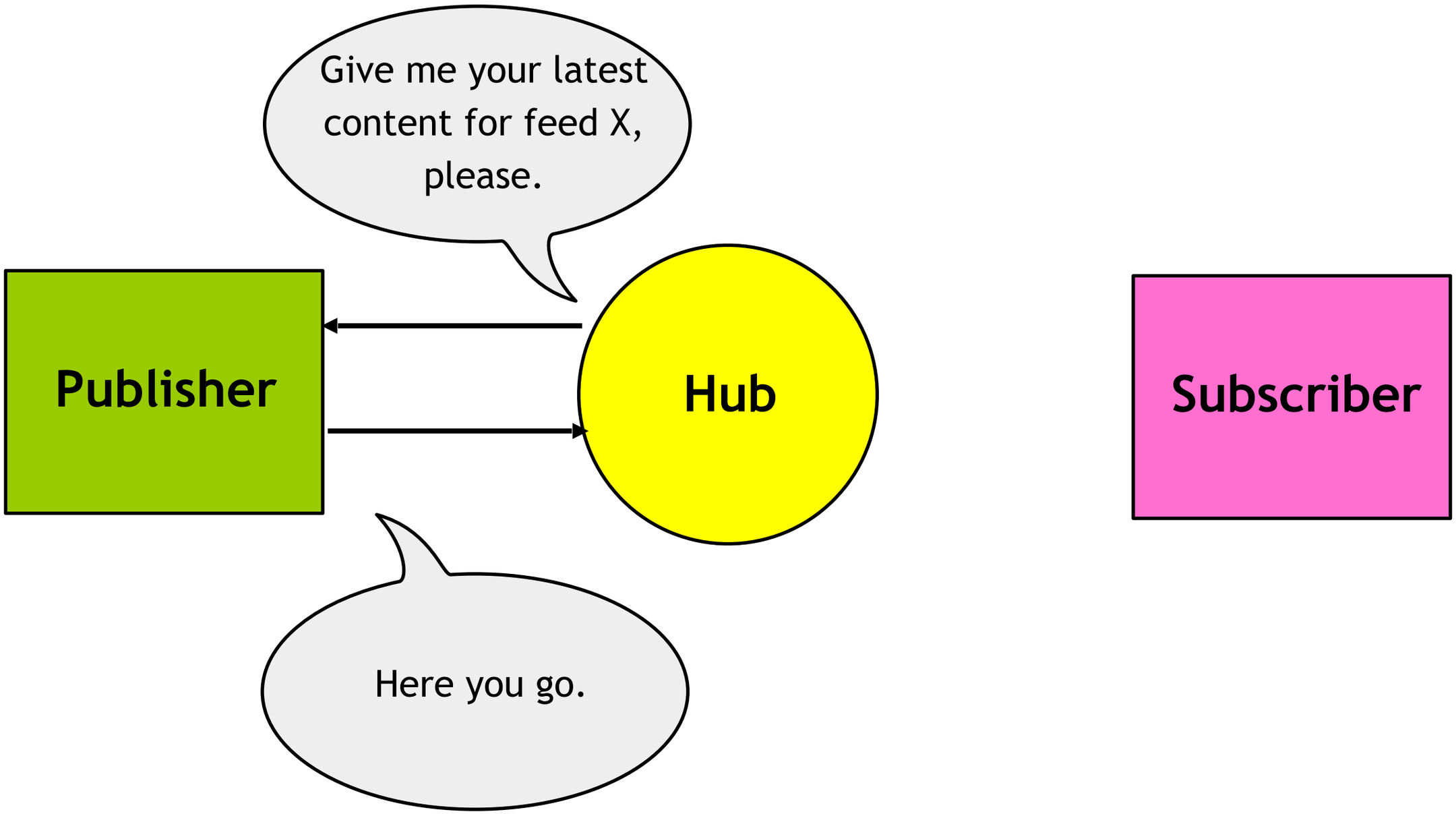
3. Hub POSTs to the endpoint URL to verify the request was authentic; Subscriber responds with confirmation to the Hub.



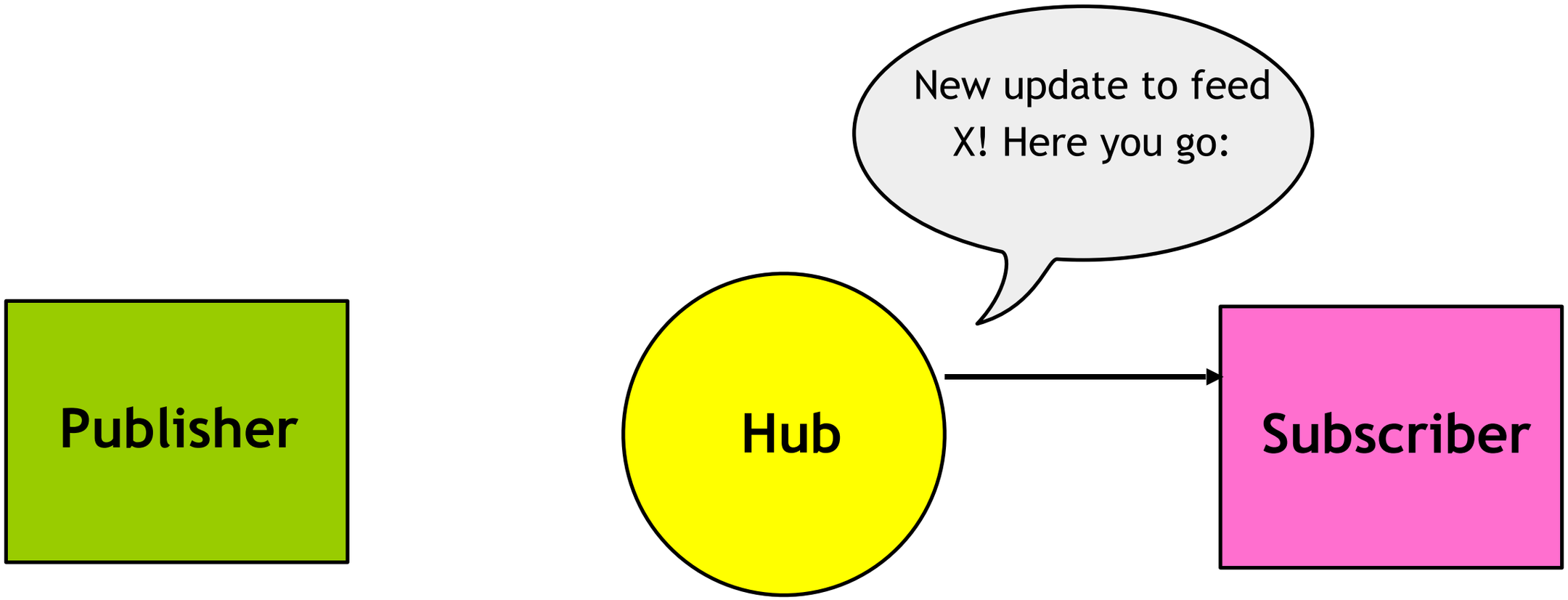
4. Publisher notifies Hub about updates by POSTing feed URLs to the Hub; Hub pulls the feed again to find new entries.



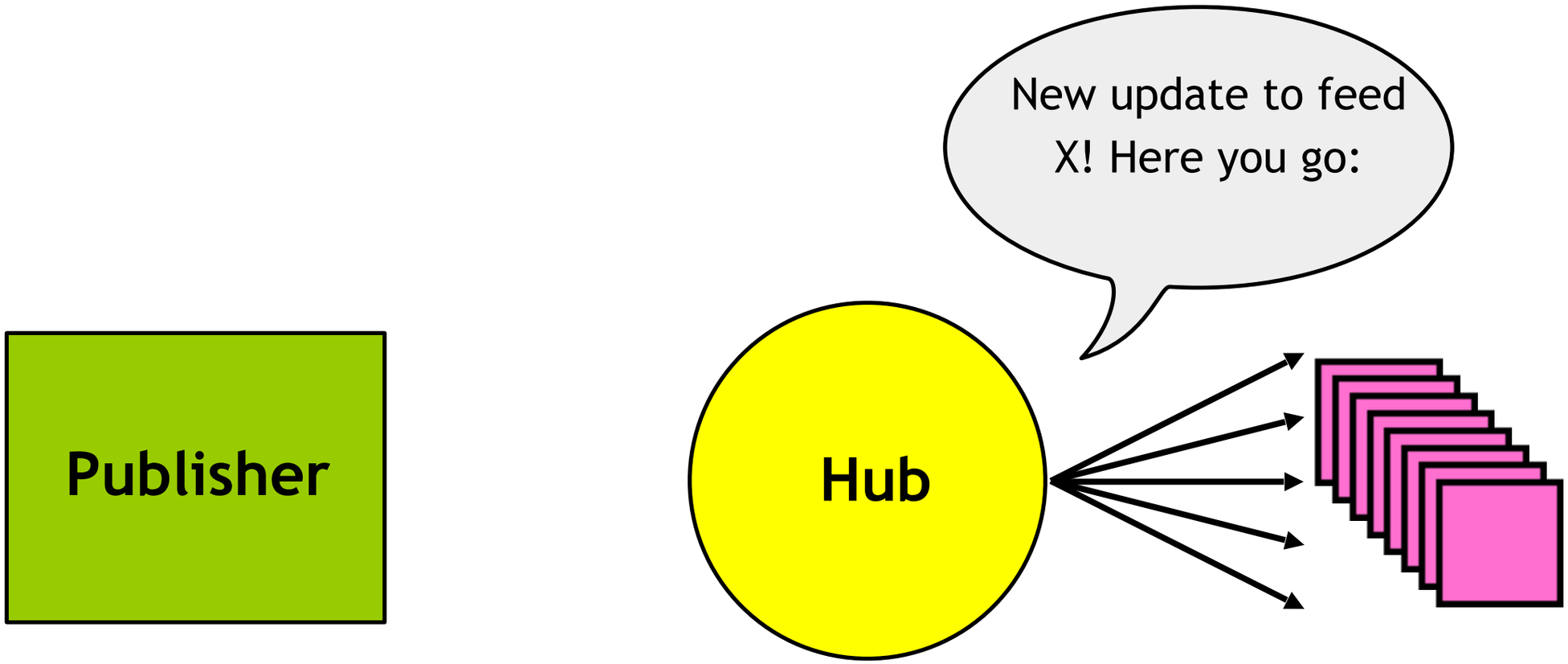
5. If the Publisher does not inform the Hub, the Hub will periodically poll the Publisher's feed for new updates.



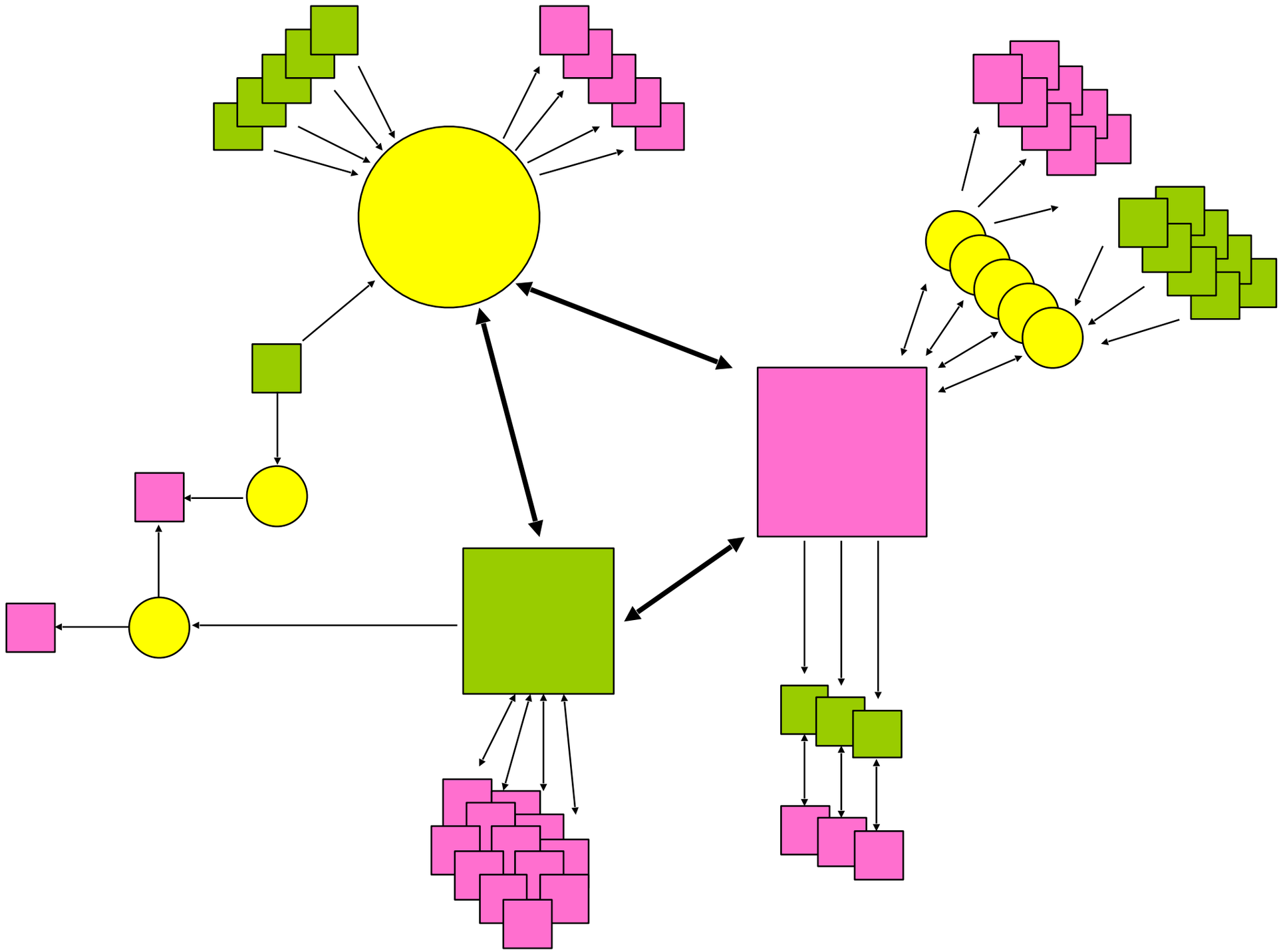
6. When Hub receives new update to feed X, it POSTs the update to the Subscriber's endpoint URL.



7. If feed X has multiple subscribers, the Hub sends updates to all of them. This reduces load on the Publisher.



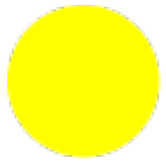
The future is distributed: There will be big hubs, many small hubs, and tons of publishers and subscribers. Publishers, subscribers, and hubs may play multiple roles.



Requirements

Publishers

- Publish Atom feeds with their content
- Include Hub forwarding information in feeds (optional)
- POST updates to Subscribers and Hubs (optional)



Hubs

- Receive subscription requests from Subscribers & verify them
- Provide endpoint URLs to which Publishers can POST updates
- Poll non-push capable Publishers to turn pull into push (optional)
- POST updates to Subscribers
- Let subscribers poll it (optional)
- Scale!

Subscribers

- Parse the feeds forwarded by the Hub
- POST subscription requests to Publishers and Hubs & confirm them
- Provide endpoint URLs to which Publishers and Hubs can post updates

Summary of terms

- Topic:
  - An HTTP [RFC2616] resource URL. The unit to which one can subscribe to changes.
- Hub ("the hub"):
  - The server (URL [RFC3986]) which implements both sides of this protocol. Any hub MAY implement its own policies on who can use it.
- Publisher:
  - An owner of a topic. Notifies the hub when the topic feed has been updated. As in almost all pubsub systems, the publisher is unaware of the subscribers, if any. Other pubsub systems might call the publisher the "source".

Summary of terms

- Subscriber:
  - An entity (person or program) that wants to be notified of changes on a topic. The subscriber must be directly network-accessible and is identified by its Subscriber Callback URL.
- Subscription:
  - A unique relation to a topic by a subscriber that indicates it should receive updates for that topic. A subscription's unique key is the tuple (Topic URL, Subscriber Callback URL). Subscriptions may (at the hub's decision) have expiration times akin to DHCP leases which must be periodically renewed.

Summary of terms

- Subscriber Callback URL:
  - The URL [RFC3986] at which a subscriber wishes to receive notifications.
- Event:
  - An event that causes updates to multiple topics. For each event that happens (e.g. "Brad posted to the Linux Community."), multiple topics could be affected (e.g. "Brad posted." and "Linux community has new post"). Publisher events cause topics to be updated and the hub looks up all subscriptions for affected topics, sending out notifications to subscribers.
- Notification:
  - A payload describing how a topic's contents have changed, or the full updated content. Depending on the topic's content type, the difference (or "delta") may be computed by the hub and sent to all subscribers.

Summary of terms

- Topic:
  - An HTTP [RFC2616] resource URL. The unit to which one can subscribe to changes.
- Hub ("the hub"):
  - The server (URL [RFC3986]) which implements both sides of this protocol. Any hub MAY implement its own policies on who can use it.
- Publisher:
  - An owner of a topic. Notifies the hub when the topic

Content Distribution

- Compare PubSubHubbub with Twitter
  - Thinking of them both as content distribution mechanisms

Twitter API

- In contrast to PuSH
  - Twitter takes control of both publish and subscribe
  - Twitter becomes a black box for multicast
  - PuSH decouples publication from subscription
    - PuSH is much more scalable
  - Not as popular

Twitter API

- There are actually three APIs
  - REST interaction
  - REST search
  - Stream-based
    - The Streaming API provides low-latency high-volume access to Tweets.

Twitter API

- You cannot make unlimited calls, follow requests, updates or direct message
  - API usage is rate limited.
  - There are limits on the number of follow requests, updates and direct messages you can make in a single day.

Twitter API

- The API is entirely HTTP-based
  - Methods to retrieve data from the Twitter API require a GET request.
  - Methods that submit, change, or destroy data require a POST.
  - API Methods that require a particular HTTP method will return an error if you do not make your request with the correct method

Twitter API

- The API is a RESTful resource
  - JSON format only as of V1.1

Twitter API

- Parameters have certain expectations
  - Some API methods take optional or requisite parameters.
  - Parameter values should be converted to UTF-8 and URL encoded.
  - The page parameter begins at 1, not 0.

Twitter API

- There are many libraries available

Twitter API

- Examples:
  - Get your data in RSS format
    - https://dev.twitter.com/docs/api/1.1/get/statuses/home_timeline

Twitter API

- Examples:
  - Streaming API
  - https://dev.twitter.com/docs/api/1.1/get/statuses/sample

Discussion

- Discussion