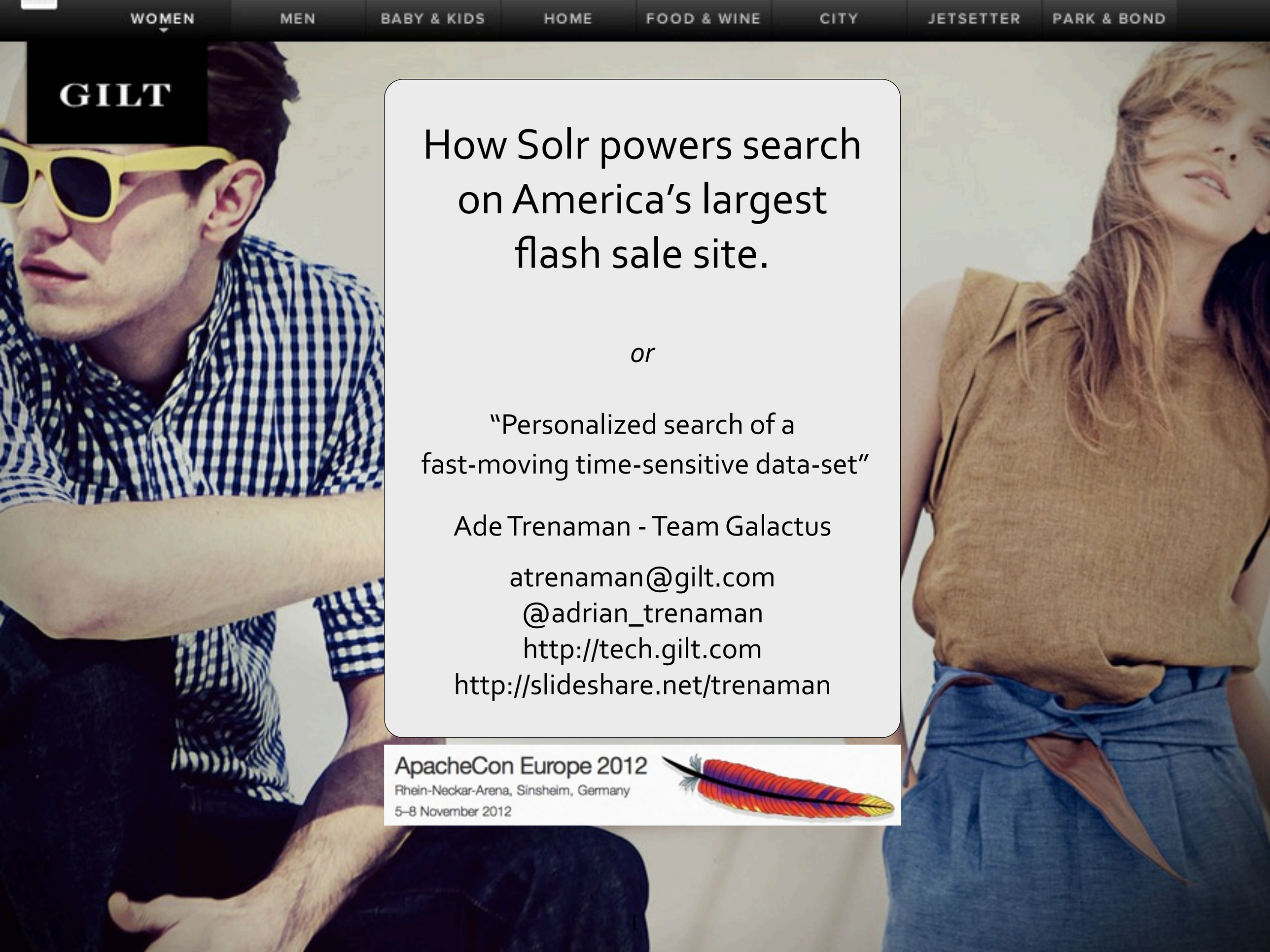


How Solr powers search on America's largest flash sale site.

or

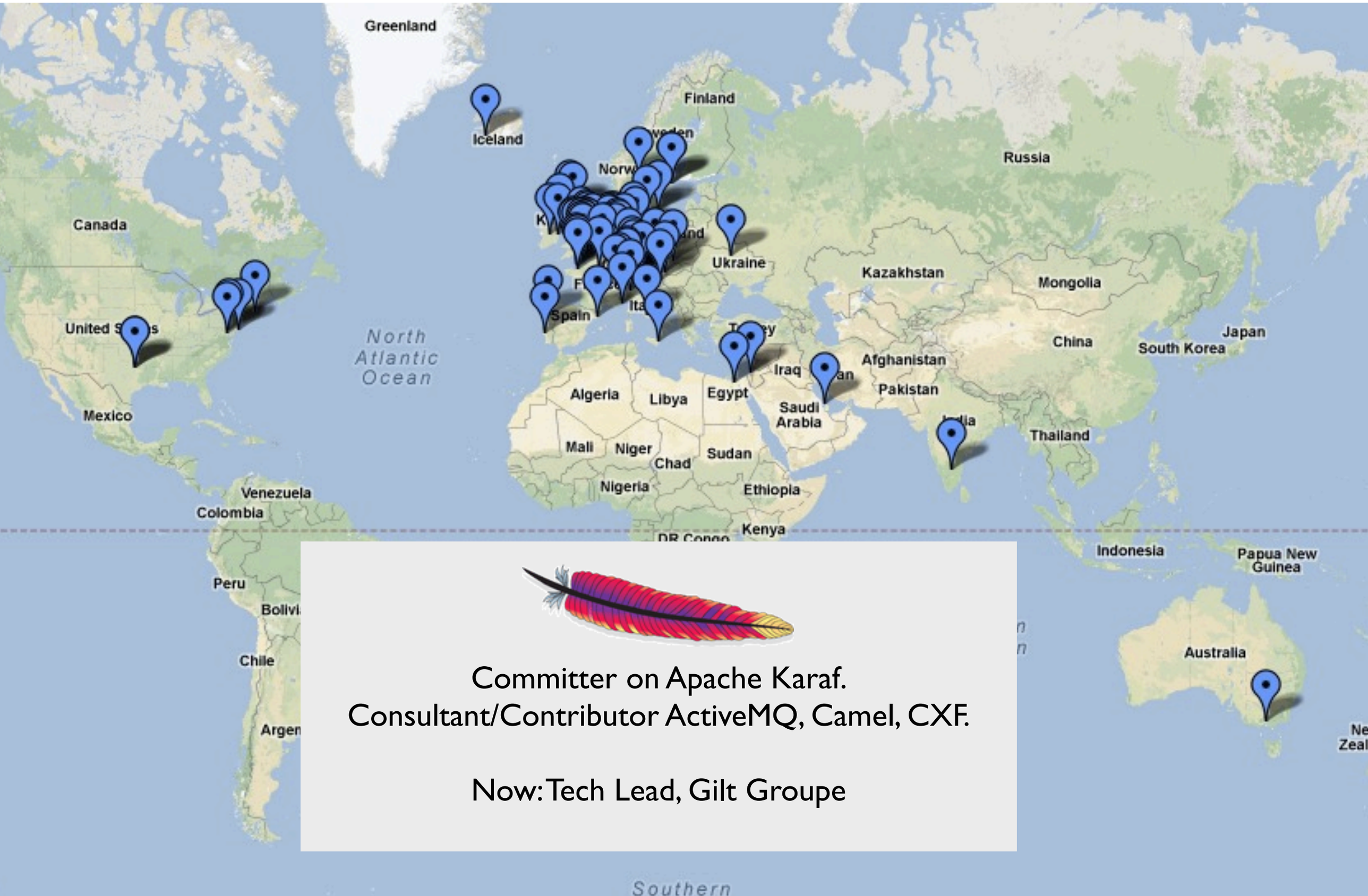
"Personalized search of a fast-moving time-sensitive data-set"

Ade Trenaman - Team Galactus

atrenaman@gilt.com
@adrian_trenaman
http://tech.gilt.com
http://slideshare.net/trenaman

ApacheCon Europe 2012
Rhein-Neckar-Arena, Sinsheim, Germany
5–8 November 2012

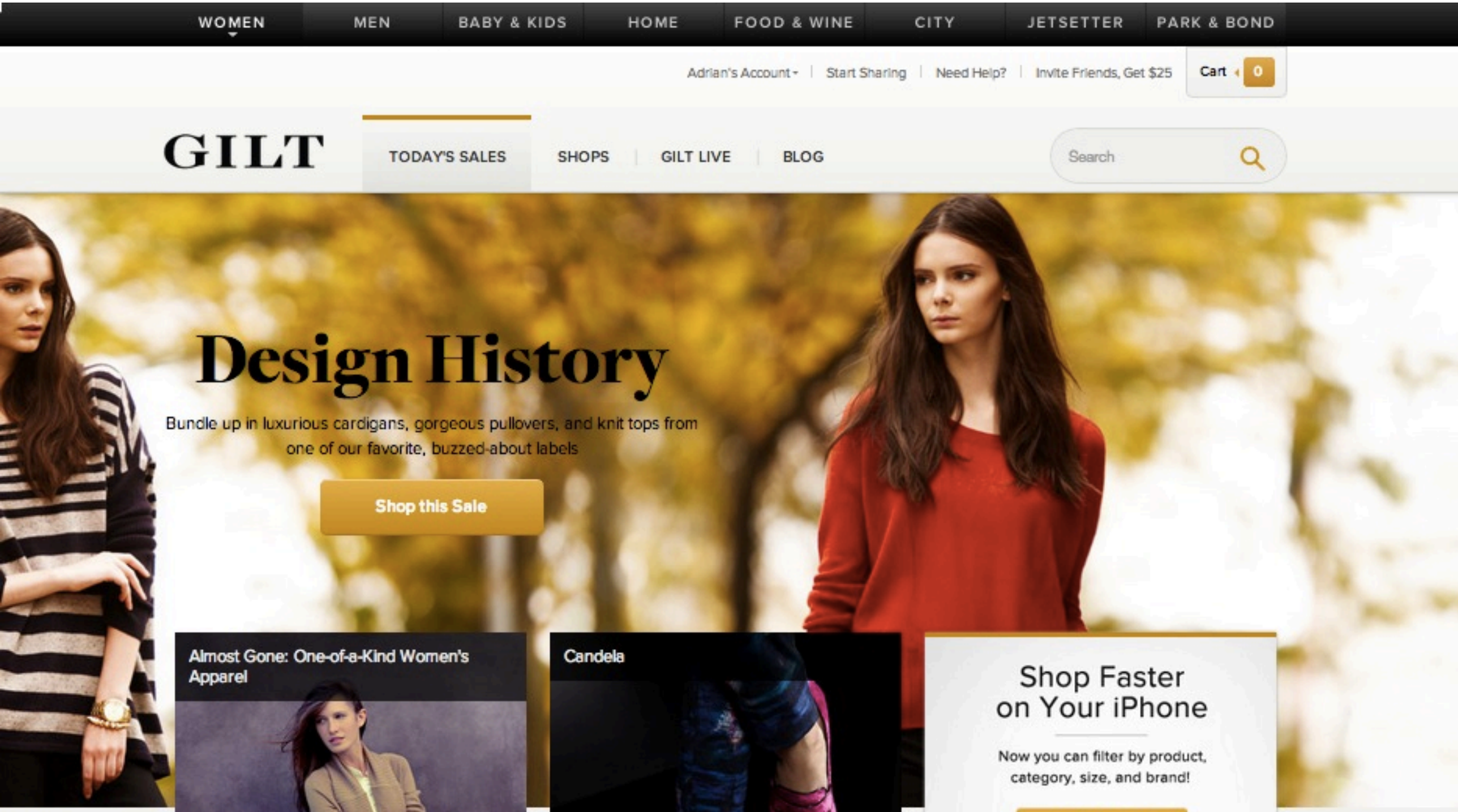
adrian.trenaman@gmail.com|@adrian_trenaman



Committer on Apache Karaf.
Consultant/Contributor ActiveMQ, Camel, CXF.

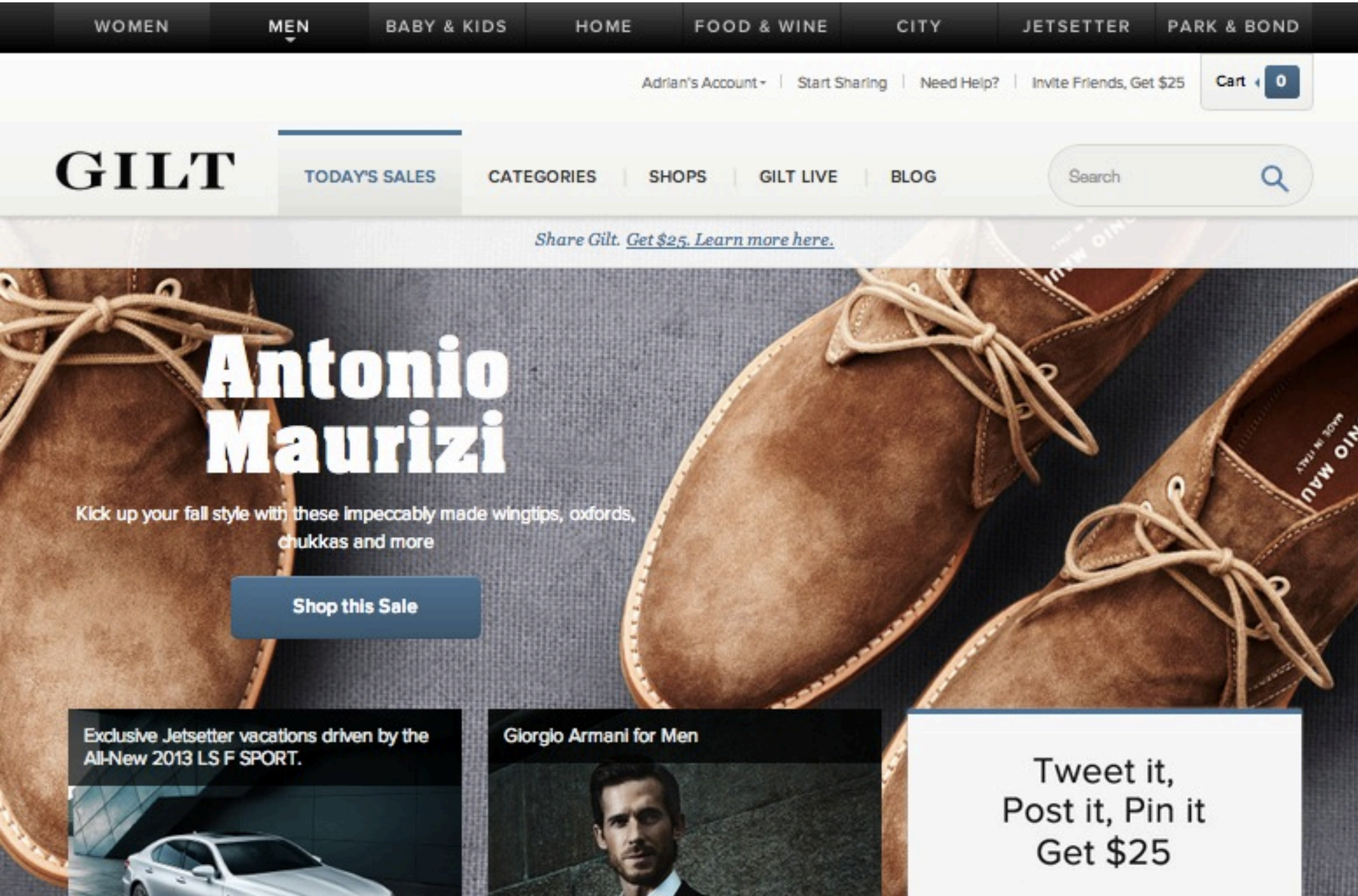
Now: Tech Lead, Gilt Groupe

Gilt: Exclusive Gorgeous Stuff on Sale at Noon!

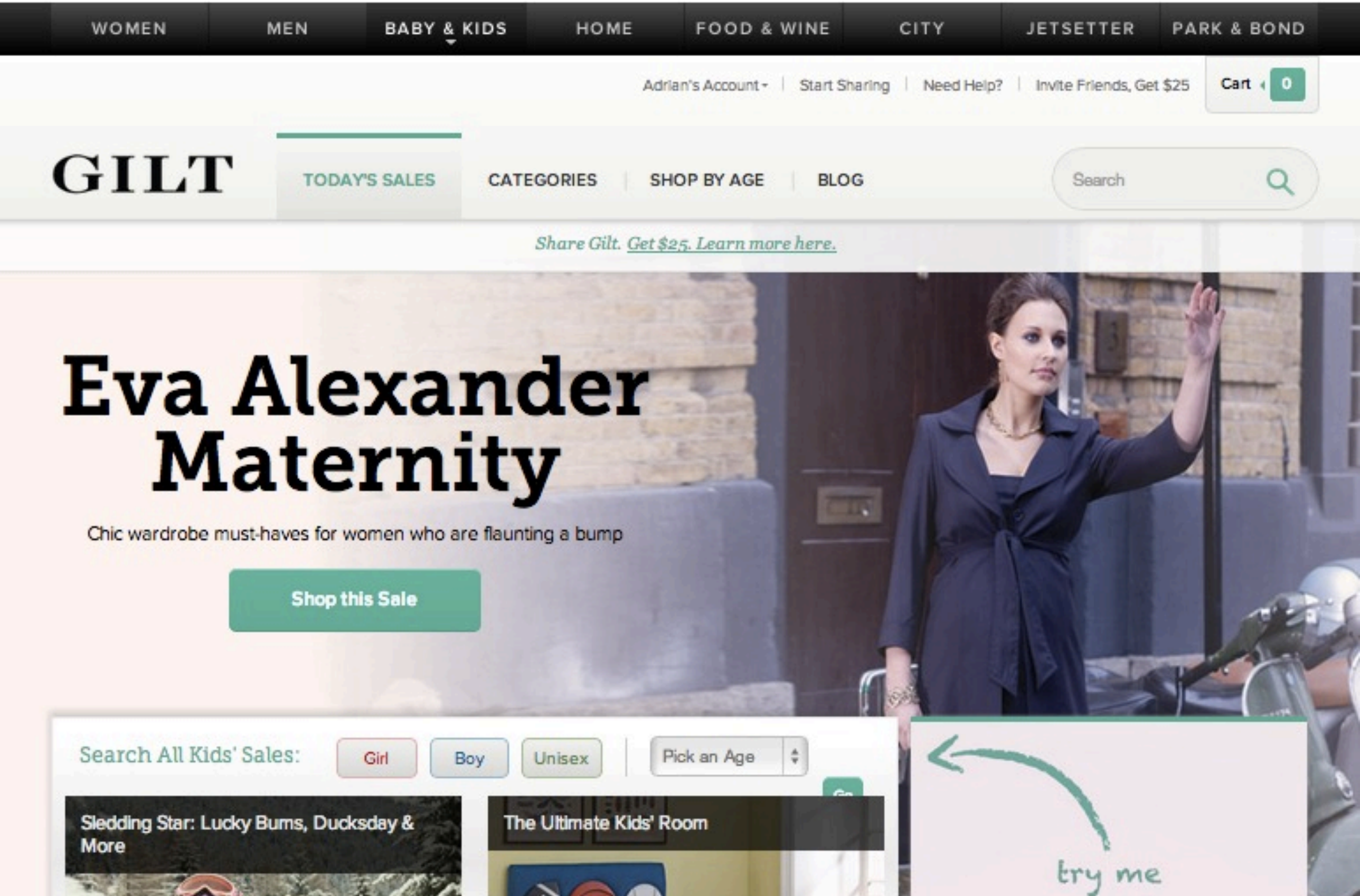


Stampede!

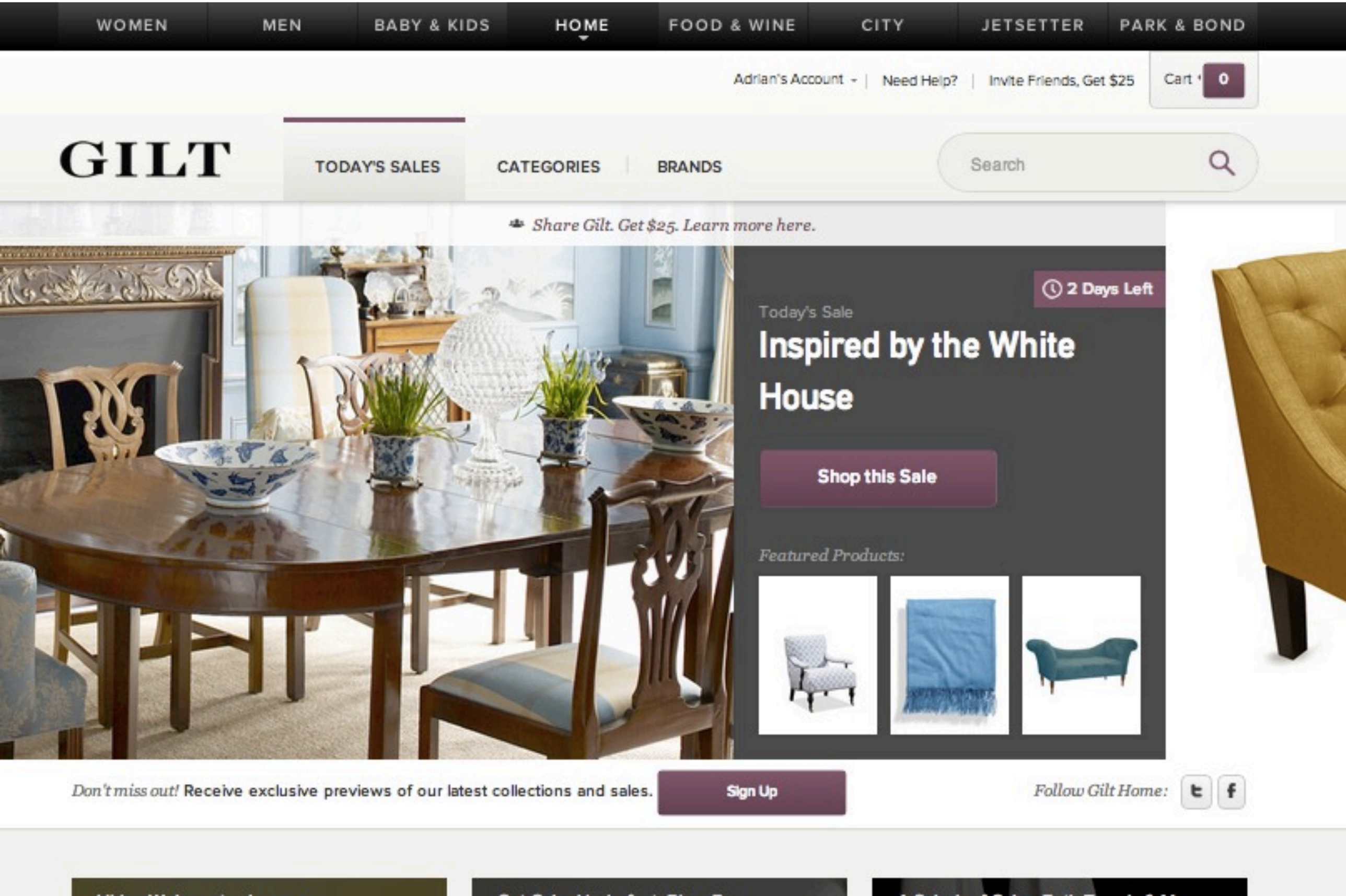
… and for guys ….


WOMEN
MEN
BABY & KIDS
HOME
FOOD & WINE
CITY
JETSETTER
PARK & BOND
Adrian's Account
Start Sharing
Need Help?
Invite Friends, Get $25
Cart 0
GILT
TODAY'S SALES
CATEGORIES
SHOPS
GILT LIVE
BLOG
Search
Share Gilt. Get $25. Learn more here.
Antonio Maurizi
Kick up your fall style with these impeccably made wingtips, oxfords, chukkas and more
Shop this Sale
Exclusive Jetsetter vacations driven by the All-New 2013 LS F SPORT.
Giorgio Armani for Men
Tweet it,
Post it, Pin it
Get $25

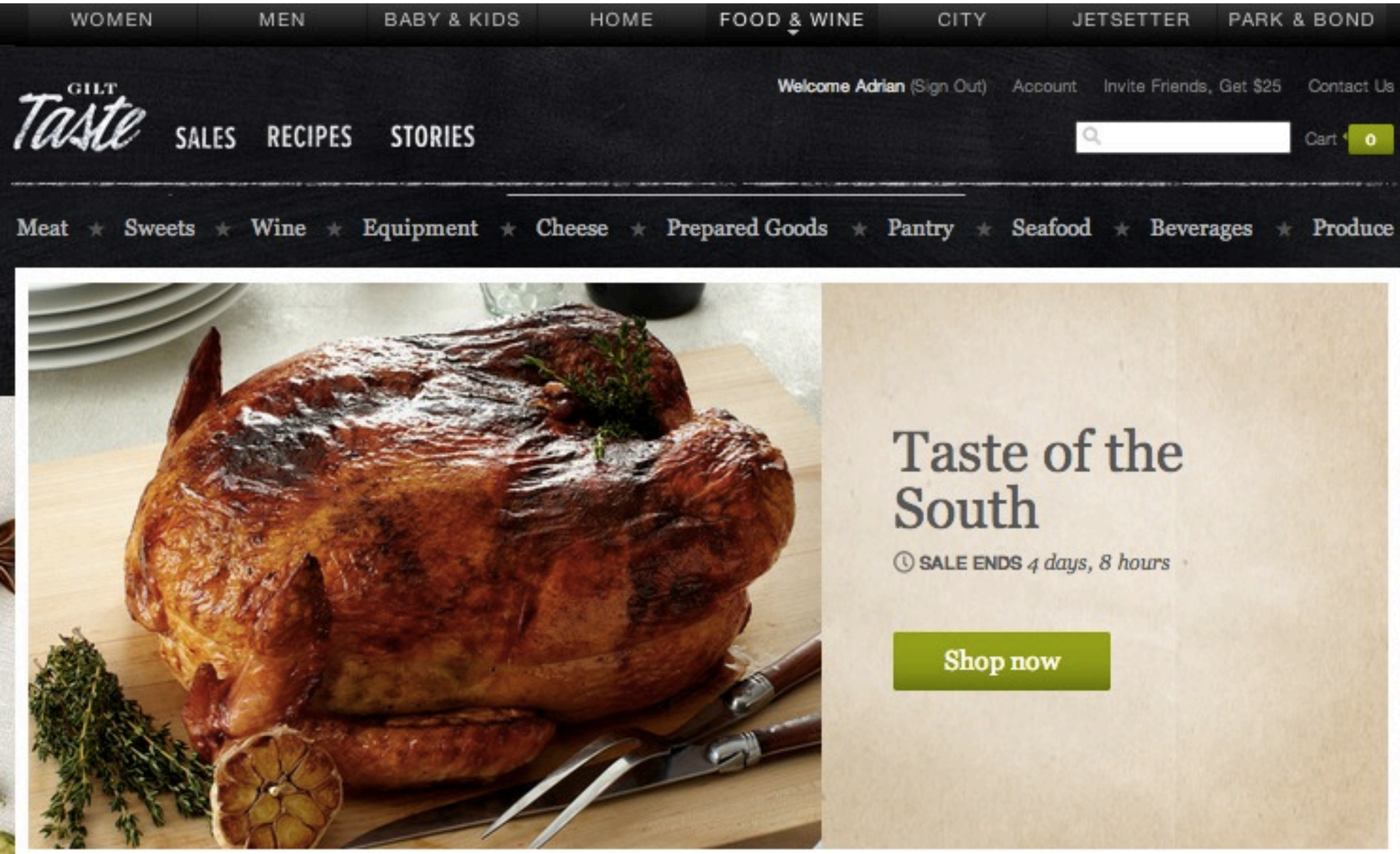
… and for baby & kids …

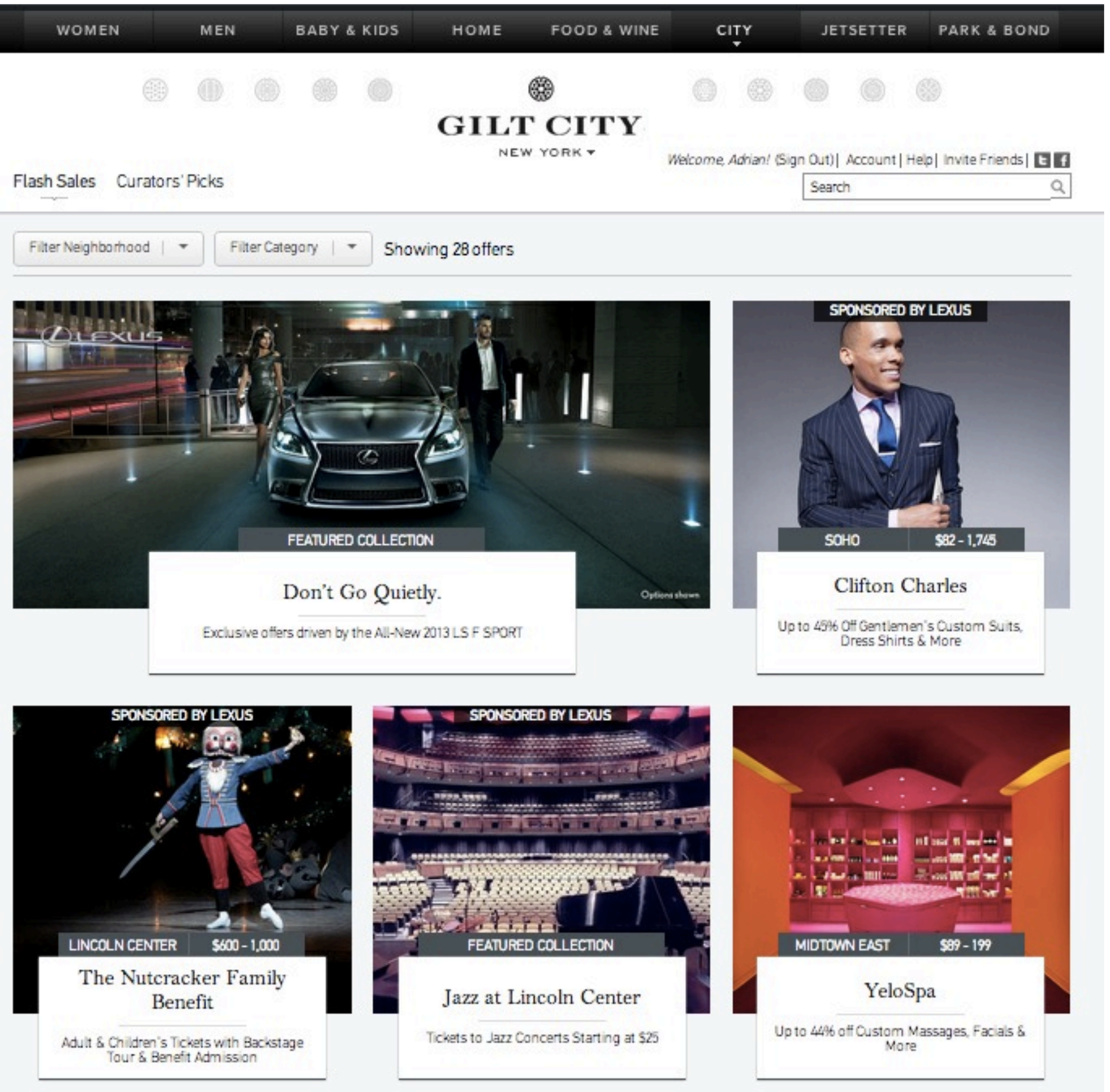


… and for your beautiful home …



… and food & fine wine …





... cool things to do in your city ...

… or somewhere else in the world.

Hotel le Bellechasse

Paris

SALE 1 day left

The Crown

United Kingdom

SALE 3 days left

Charleston Place - S.C.

South Carolina

SALE 5 days left

An Uptown NYC Mystery Hotel

The Gilt noon 'attack of self denial'

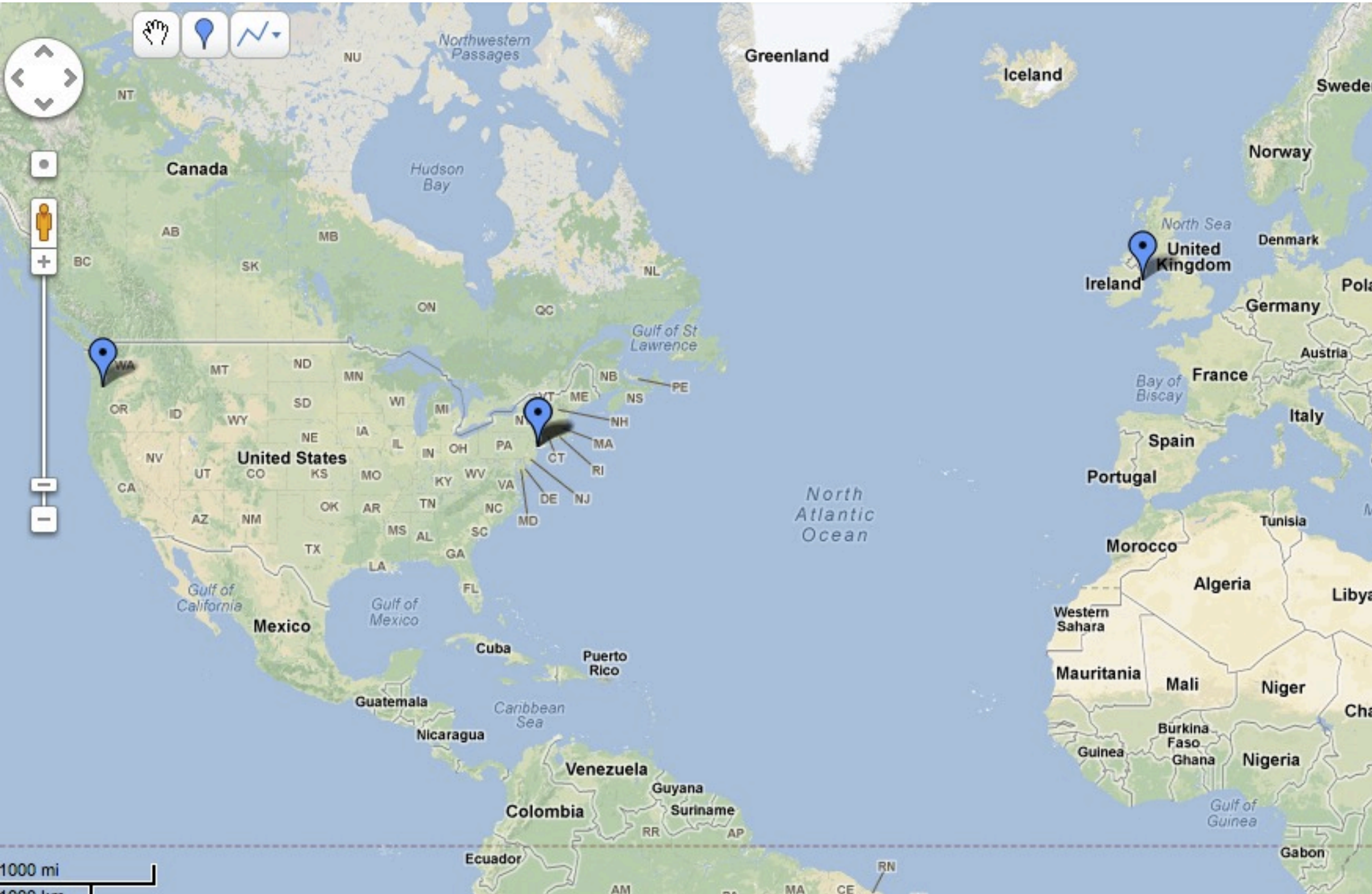
- Email & Mobile notification of today's sales goes to members at 11:45 EST…
  - … Sales go live at Noon EST.
- Stampede.

http://gilt.com/apacheconeu2012

$25

... our treat to you.

PDX, NYC, DUB - tech.gilt.com



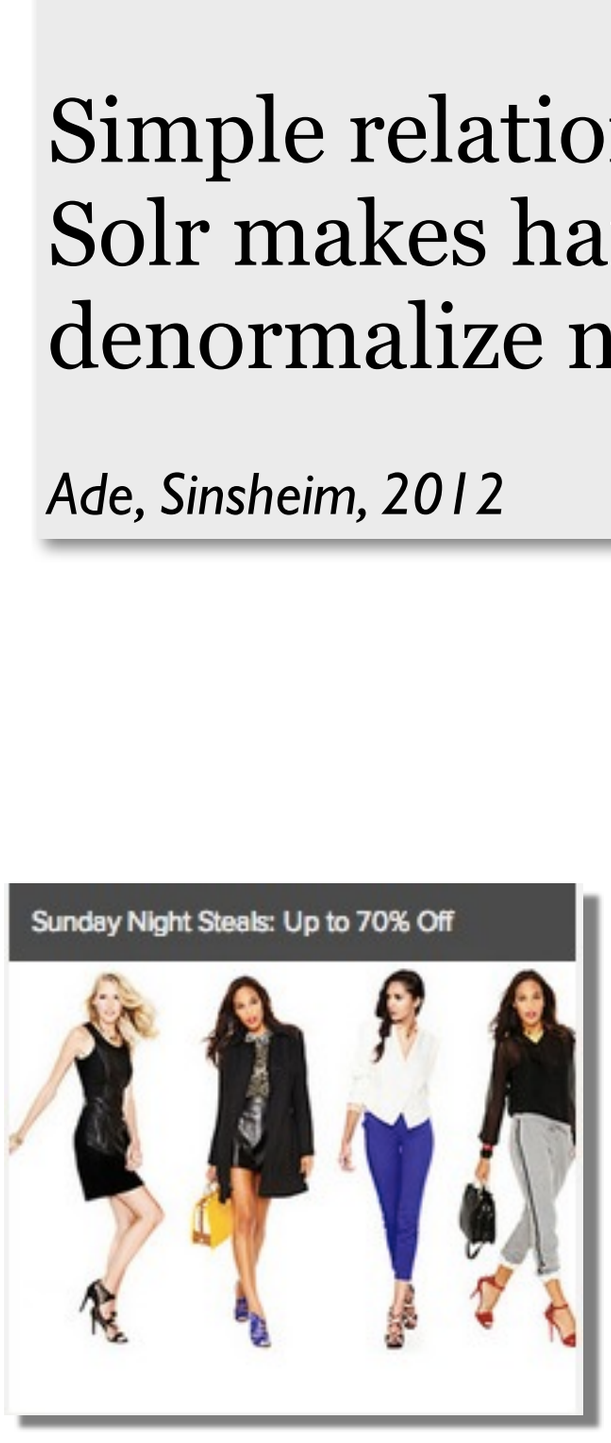
The problem

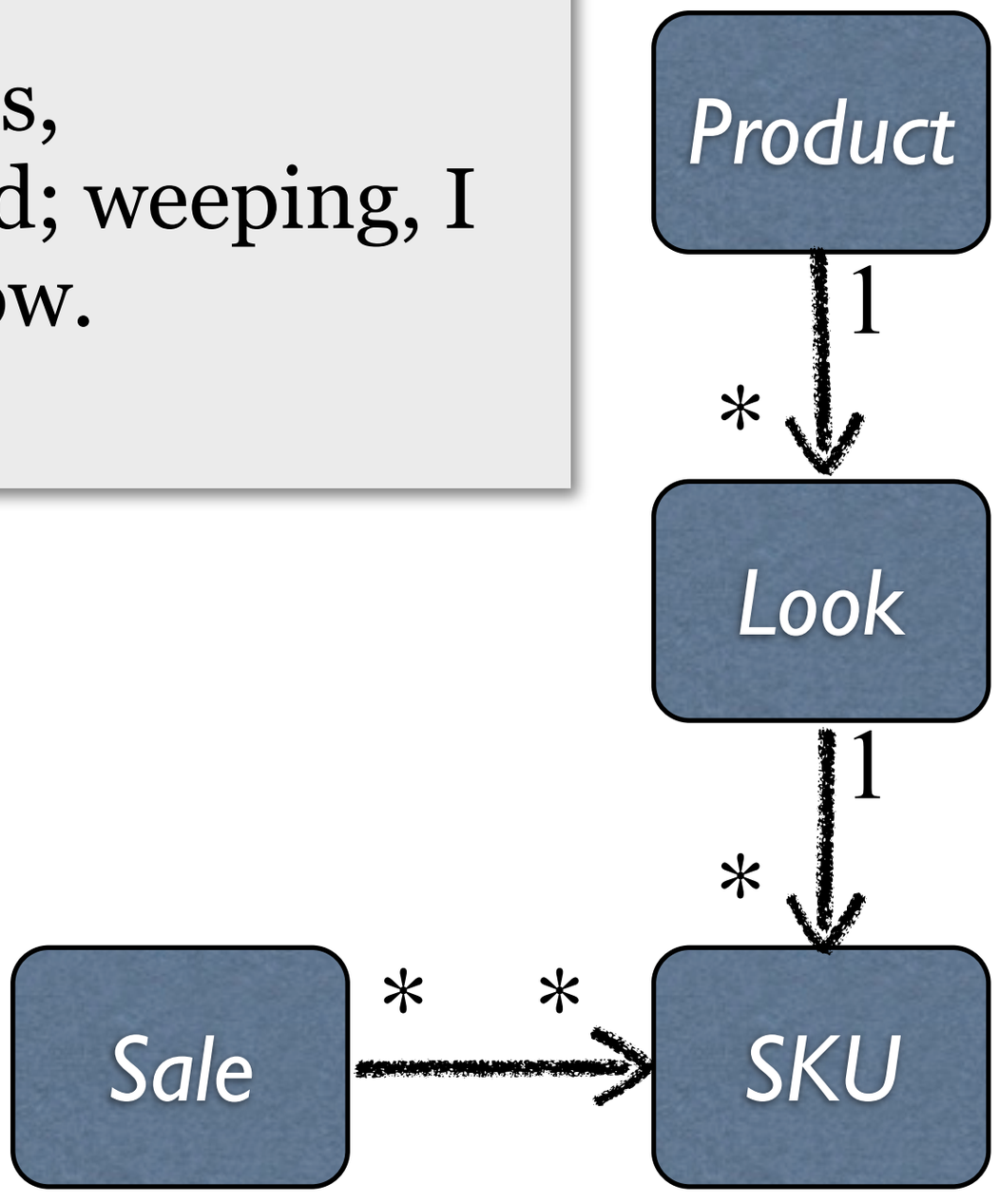
Gilt Data Model 101

A Haiku:

Simple relations,
Solr makes hard; weeping, I
denormalize now.

Ade, Sinsheim, 2012



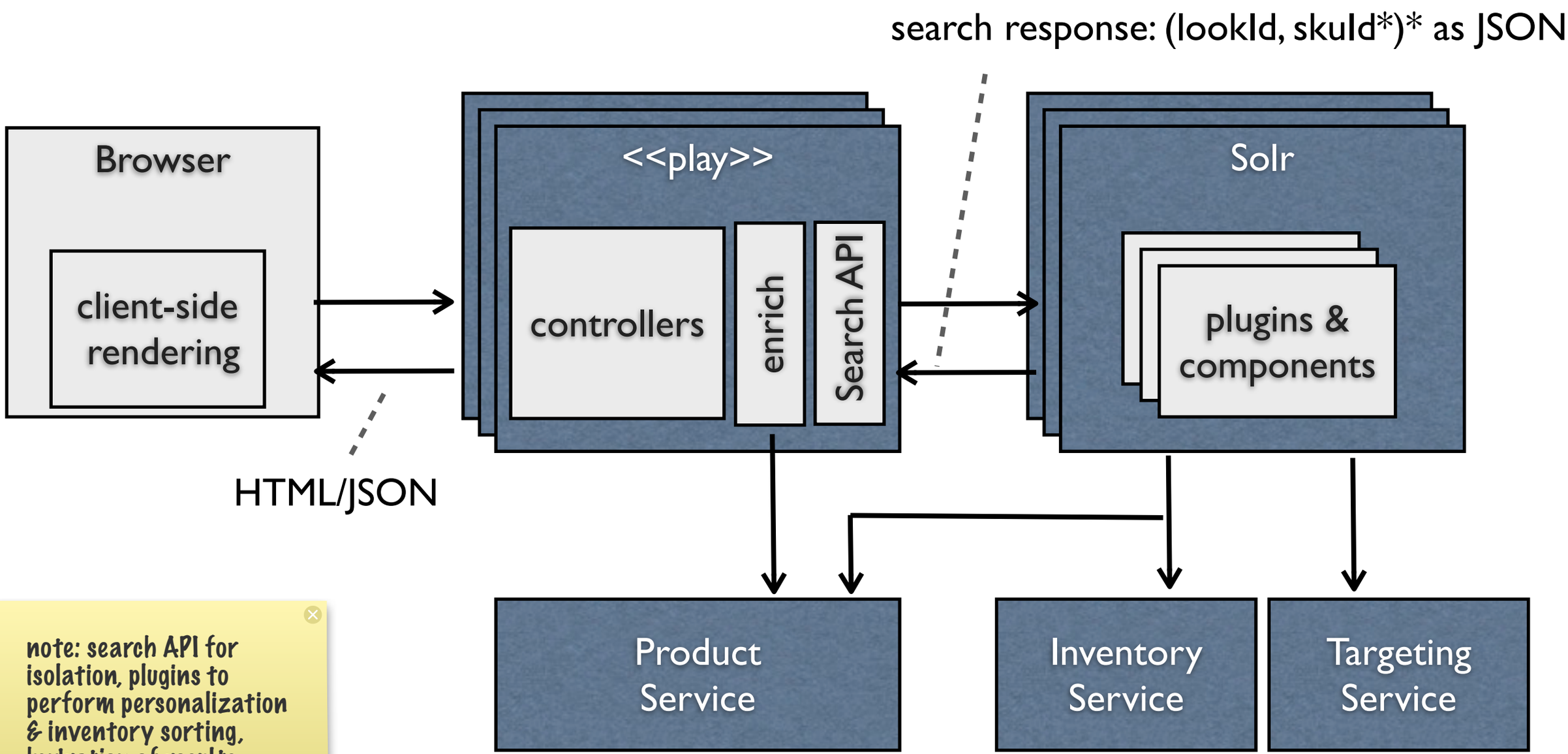


Silk Charmeuse Wrap Dress

... in white

... in size 8

QED Architecture Pattern: Query, Enrich, Deliver

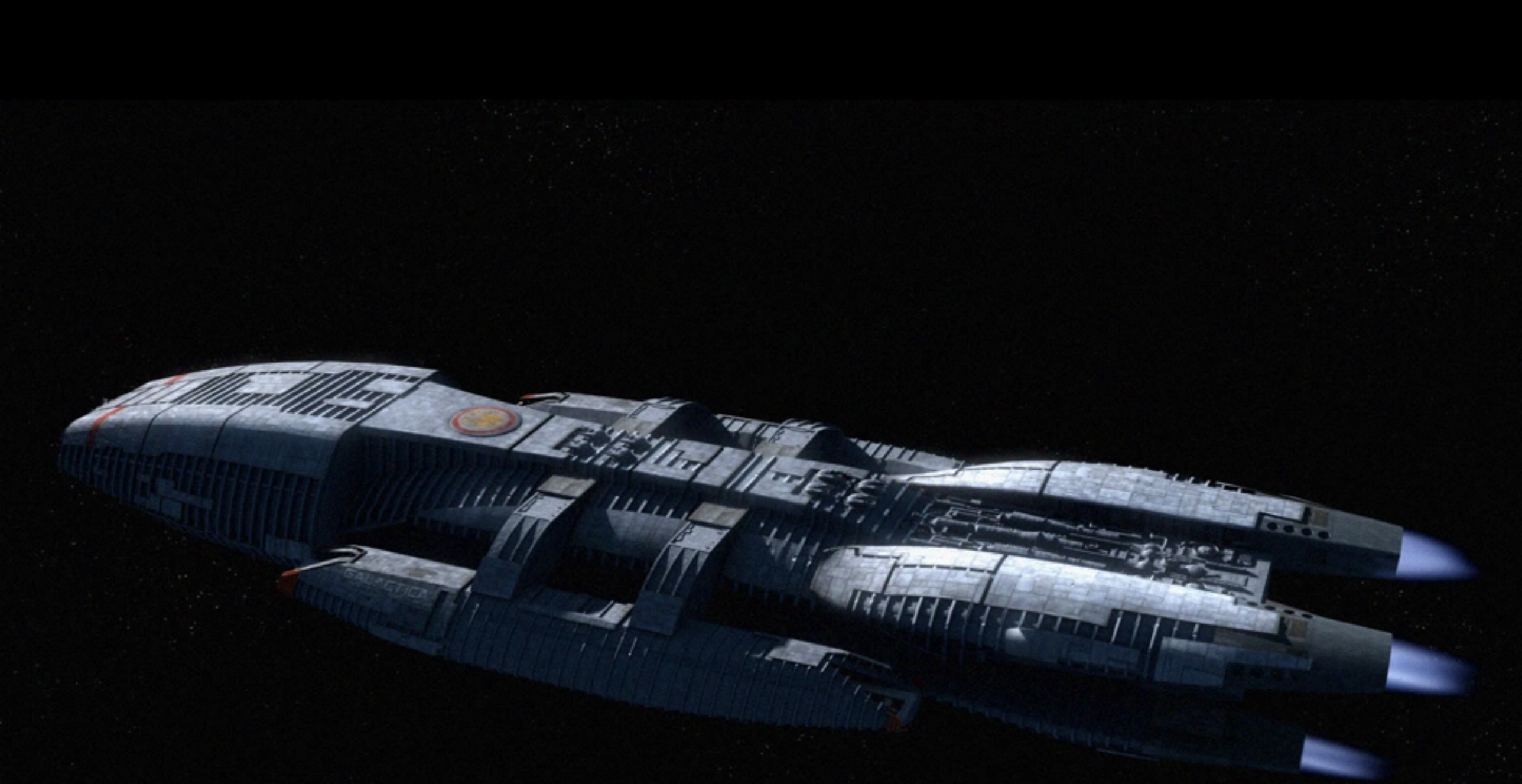


note: search API for isolation, plugins to perform personalization & inventory sorting, hydration of results

Motivating principle: "Do One Thing Well"

Search MVP

- Provide search listings of ‘product looks’
  - Move sold-out inventory to bottom in realtime
  - Facet by Size / Color / Brand / Category
  - Search by store (Home, Women’s, …)
  - Provide targeted search results
  - Respect sale start / end time in listings
  - Auto-suggest
  - Auto-complete



Search Space

Search space: what to index?

Taxonomy

Brand, Product Name

Price

Color

Inventory Status

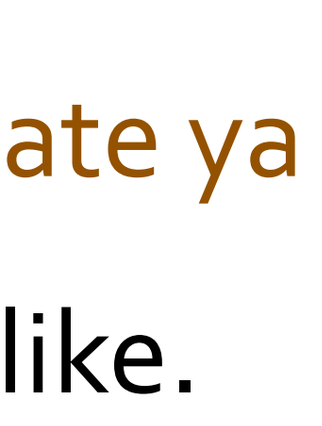
SKU Attributes / Material / Origin

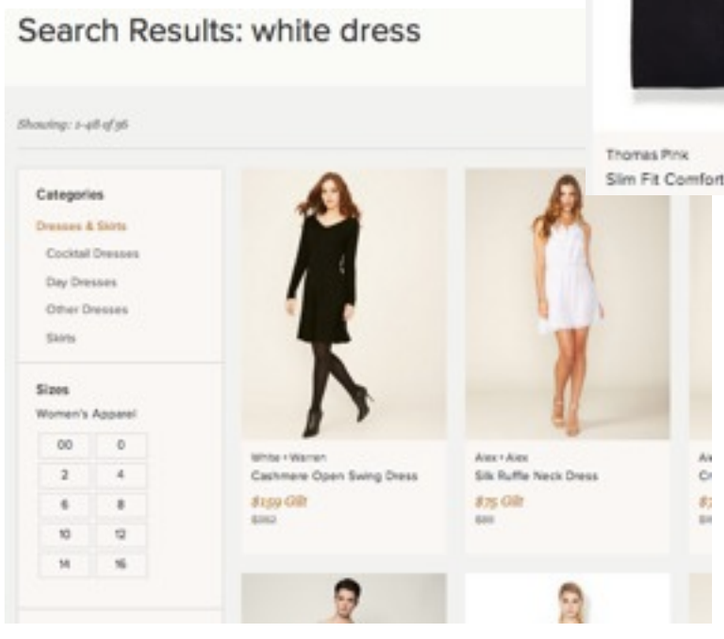
Description

Data, data, data gonna hate ya, hate ya, hate ya

- Not all product data is as clean as you'd like.
  - Description contains distracting tokens (**'also available in blue, black (patent), green and leather'**)
  - Colors are often poorly named: **'blk', 'black', 'priest', 'black / white', 'nite', 'night', '100', 'multi', 'patent'**.
  - Brand names can mislead: **'Thomas *Pink* Shirts',**

    'White + Warren Black Dress'





- Trust nothing.
  Try everything.
  - Be ready for surprises…

Example: synonym leakage

- Naive and excited, we configured the Princeton Synonym database with Solr :)
  - Search for **"black dress"** yields **"Jet Set Tote" :(**
  - Learning: don't blindly rely on synonyms.

huh?

Doh!

*(**black** → 'jet black' → **jet**) (**dress** → **set**)*

Index by looks or by SKU?

- **Turmoil:** we list *looks*, but filter by *SKU attributes* (e.g. size).
- **Bad Idea:** index looks, with size as a multi-value field:

```
product_look_sizes = "S", "M", "L", "XL"
```

- … filtering with `product_look_sizes:M` might return a product that has inventory for S, L, XL but no inventory for "M". *Really Bad Experience*.
- **Better idea: *denormalize***. Index by SKU with a single-valued field size field:

```
sku_size = "M"
```

- … now, when someone indexes by "M", they only SKUs in medium. Bazinga!

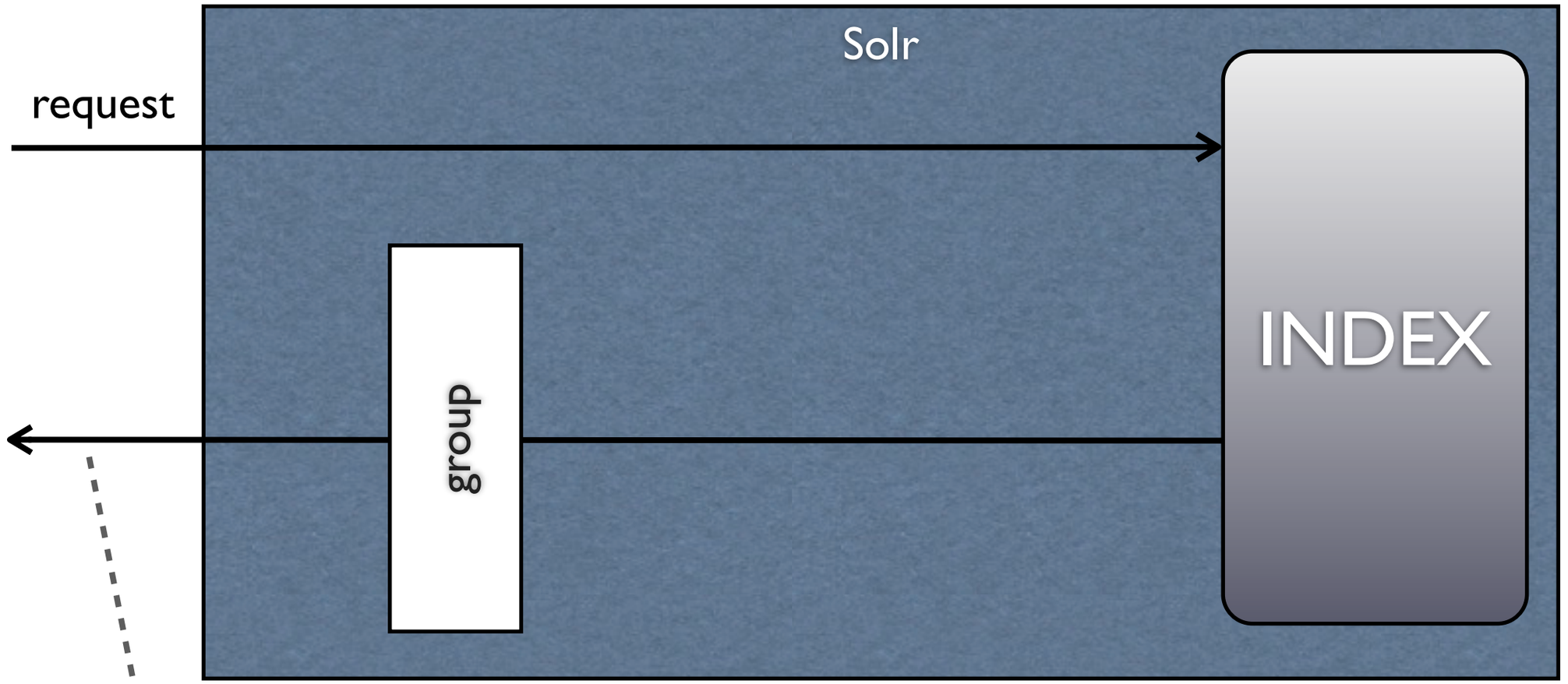
Index by looks or by SKU? (cont')

- But we list looks, not SKUS!!
  - The last thing a customer wants is to see 8 pictures of the same dress in 8 different sizes
- Group by 'product look ID' using Solr grouping: works a treat!

```
<str name="group">          true          </str>
<str name="group.field">    sku_look_id_s </str>
<str name="group.limit">    -1            </str>
<str name="group.ngroups"> true           </str>
```

- Initial worries about performance are (so-far) unfounded.
  "Premature optimization is the root of all evil."
- We -may- change to block-join queries when we move to Solr4.

Solr Internals & Extensions I



as JSON

Realtime Inventory Ordering

Real-time inventory ordering

- **Crucial:** sold-out looks should move to bottom of list; i.e. sort by 'inventory status'.
- Given: an asynchronously updated caching 'inventory status' API.

```
InventoryStatus status = getInventory(sku)
```

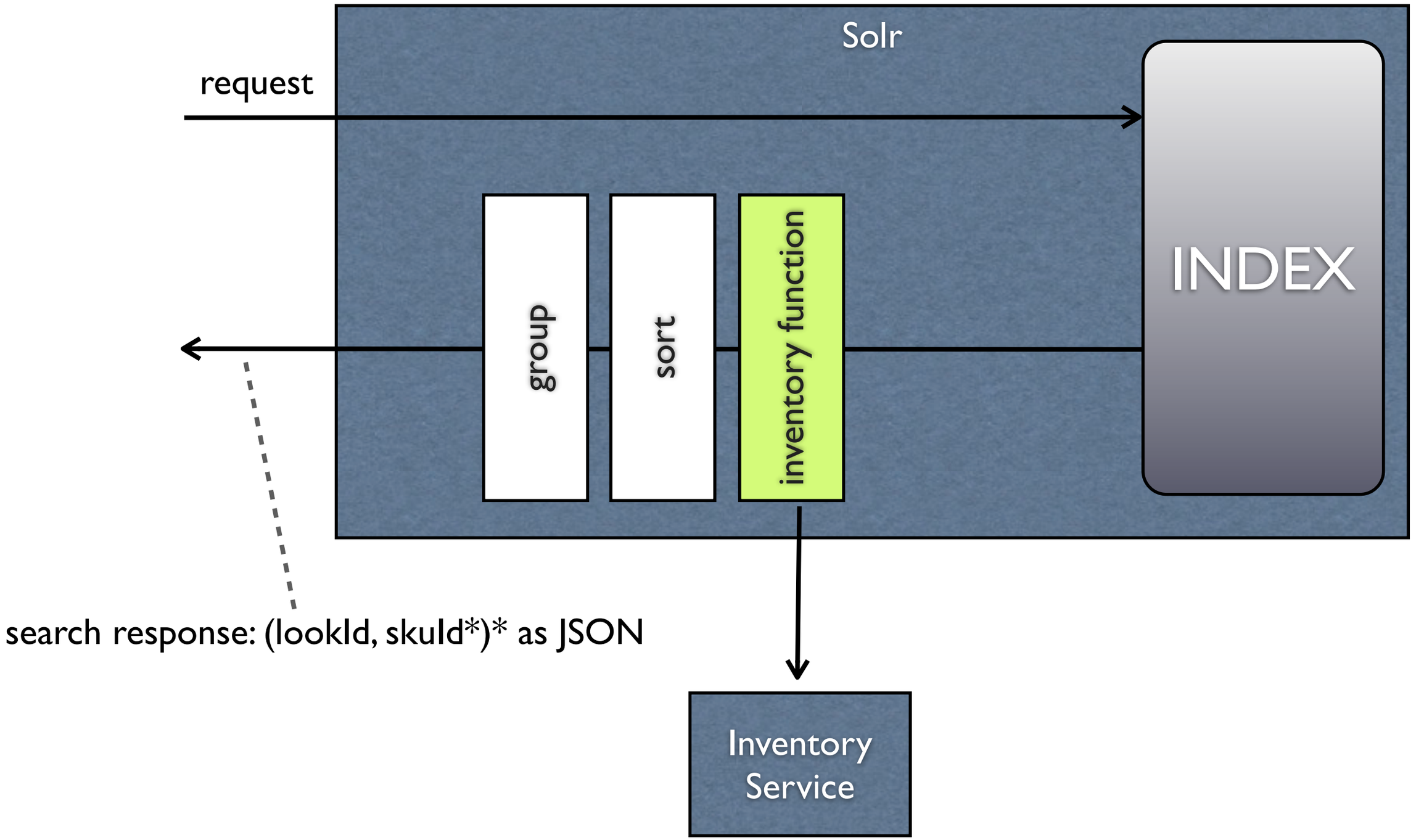
- … we created a custom `inventory()` function in Solr that assigns a numeric value (0/1) to each SKU in a result [O(n)]
- Queries can now be sorted using:

```
order by inventory(sku), score
```

Real-time inventory ordering (cont')

```
public class InventoryFunction extends ValueSource {
  public DocValues getValues(...) throws IOException {
    return new DocValues() {

      private float inventory(int doc) {
        InventoryStatus skuInventoryStatus =
          inventory.getSkuInventory(getSkuId(doc)).getStatus();
        return (skuInventoryStatus == InventoryStatus.SOLD_OUT) ? 0 : 1;
      }

      public float floatVal(int doc) { return inventory(doc); }
      public double doubleVal(int doc) { return inventory(doc); }
      : : :
    };
  }
}
```

Solr Internals & Extensions II



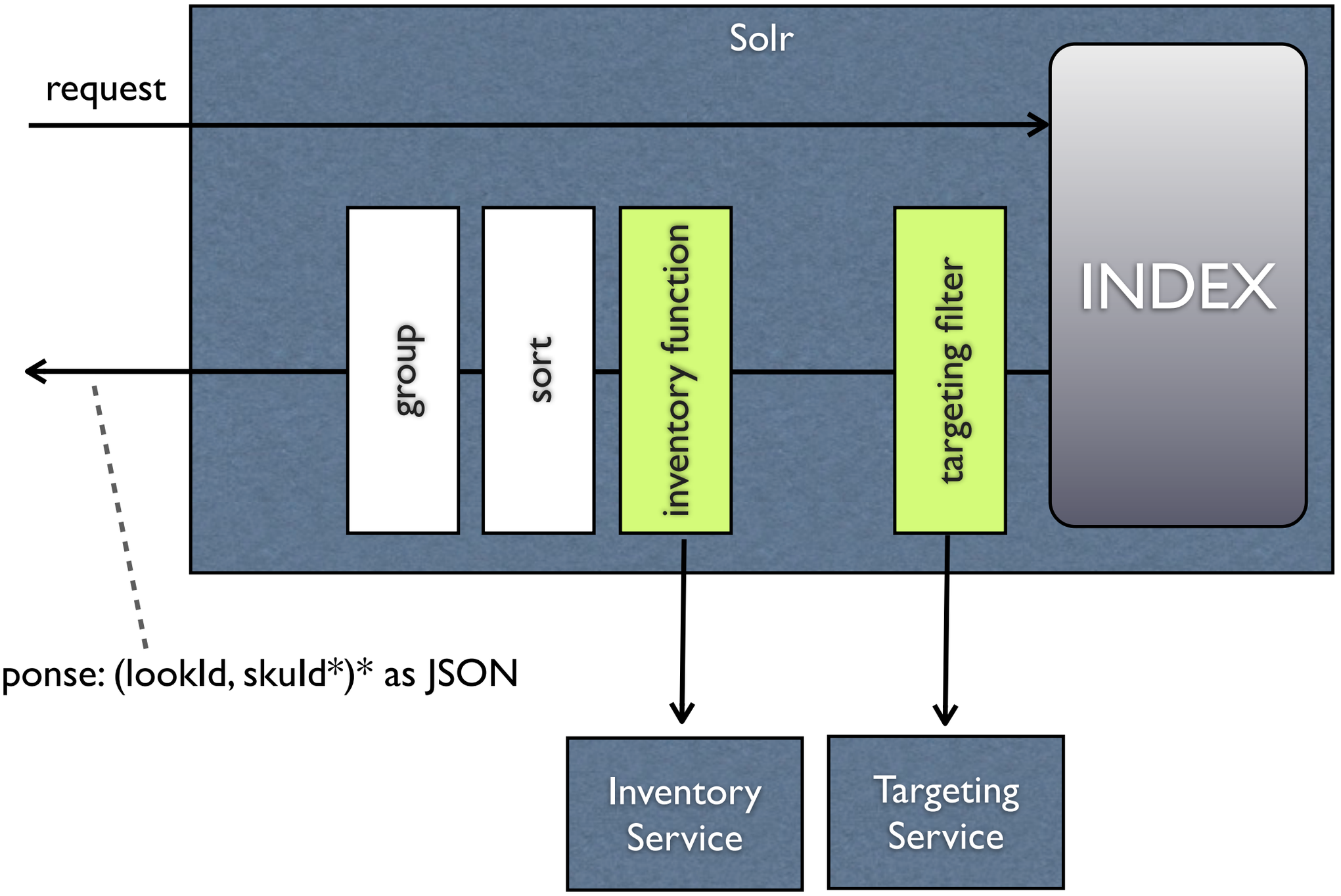
Targeting & Personalization

Targeting

- Sale-targeting is an important tool for Gilt merch:
  - Groups (*Noir* members, employees, …), geo-IP, weather, inclusion/exclusion lists
- Search listings and auto-suggest *must* respect sale targeting. Solution:
  - Store the saleId that the SKU is available in the index
  - Pass the user's GUID to Solr
  - Filter results using a custom filter

```
if (! allowed(saleId, userGuid)) {
  // remove result from listing
}
```

Solr Internals & Extensions III



Timing is Everything

Our index is time-sensitive

- Sales go live at Noon EST:
  - ***Do not*** want products to be searchable if the sale is not yet active.
  - ***Do*** want products to be searchable at exactly Noon EST
- We index Solr every n minutes (n == 15)
- Need to encode a sense of time in the index.

Time-sensitive search

- 1st idea: index a SKUs start/end availability and then filter on that.
- Use Solr's 'dynamic fields':

```
for each store (men / women / home / kids / ...) {
  // get the nearest upcoming availbility window & index it.
  sku_availability_start_<store> = startTime
  sku_availability_end_<store> = endTime
  sku_availability_sale_<store> = saleId
}
```

- Then, naively filter using (see next slide for improvement):

```
fq=+sku_availability_start_<store>:[* TO NOW]
   +sku_availability_end_<store>:[NOW+1HOUR TO *]
```

All praise and thanks to the chump.

- Filtering by time for every request is expensive :(
- Posed question to The Chump at Lucene Revolution 2012.
- Solution: rounding time bounds to nearest hour / minute means we get a cached query filter.

`[* TO NOW]` → `[* TO **NOW/HOUR**]`

- Super Chump. Chump is Champ. Chump-tastic. Chump-nominal. Awe-chump.

Time-sensitive search (cont')

- e.g. Consider a SKU available in women & home:

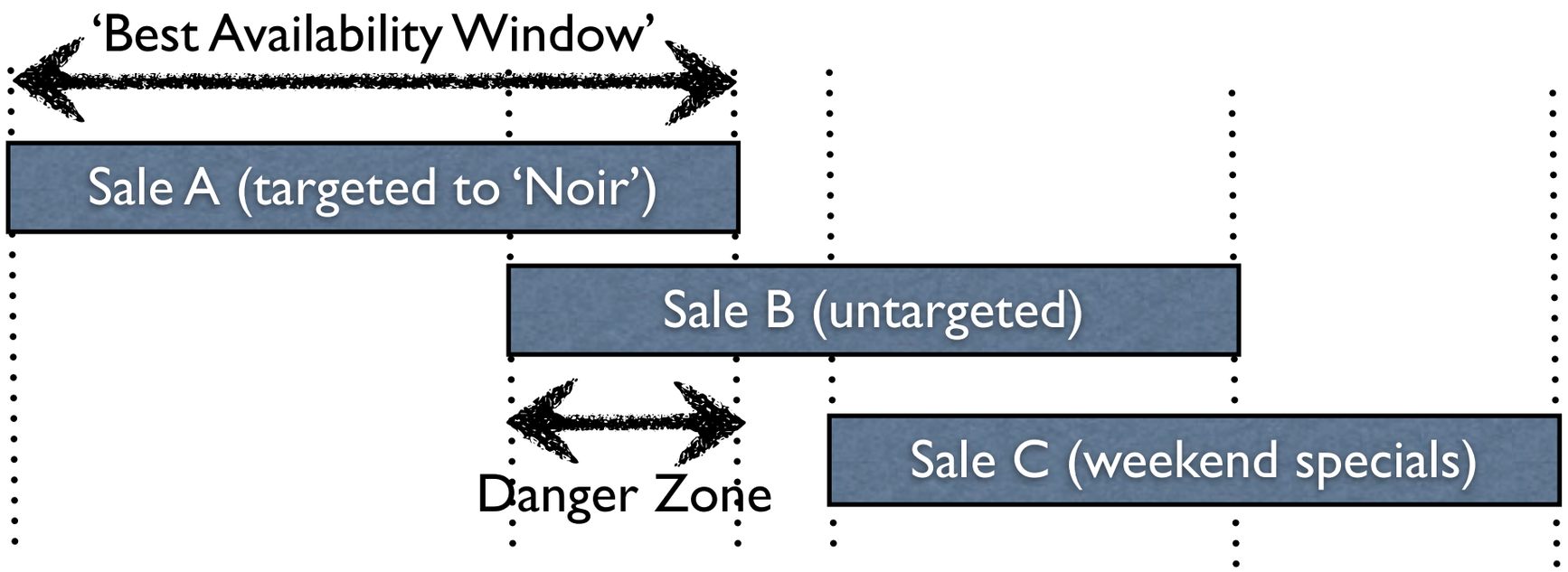
```
sku_availability_start_home = 32141232 // Time

sku_availability_end_home = 32161232 // Time

sku_availability_sale_home = 12121221 // SaleID

sku_availability_start_women = 32141232 // Time

sku_availability_end_women = 32161232 // Time

sku_availability_sale_women = 223423424 // SaleID
```

- Now, can search in Home store using:

```
fq=+sku_availability_start_home:[* TO NOW/HOUR]
+sku_availability_end_home:[NOW/HOUR+1HOUR TO *]
```

Problems in the space-time continuum (cont’)

- At index time, we used a heuristic to determine the 'best availability window' when a SKU is in 2 sales in the same store
  - “Earliest active window, or soonest starting window”: some windows ignored
  - Targeting problems: might remove a SKU in a restricted sale, even if the SKU is visible in another Sale :(



Problems in the space-time continuum (cont')

- Also, some sales are restricted to the *channel* the product is in!
  - 'Channels': e.g. mobile, iPad, …
- Our 'best availability window' code was already creaking; Introducing more dynamic fields (channel x store) *stank*.

```
sku_availability_start_web_home = 32141232 // Time

sku_availability_end_web_home = 32161232 // Time

sku_availability_sale_web_home = 12121221 // SaleID

sku_availability_start_ipad_home = 32141232 // Time

sku_availability_end_ipad_home = 32161232 // Time

sku_availability_sale_ipad_home = 12121221 // SaleID

sku_availability_start_mobile_home = 32141232 // Time

: : : :
```

Eureka!

- Stop indexing SKUs; instead: index 'the availability of a SKU at a moment in time'

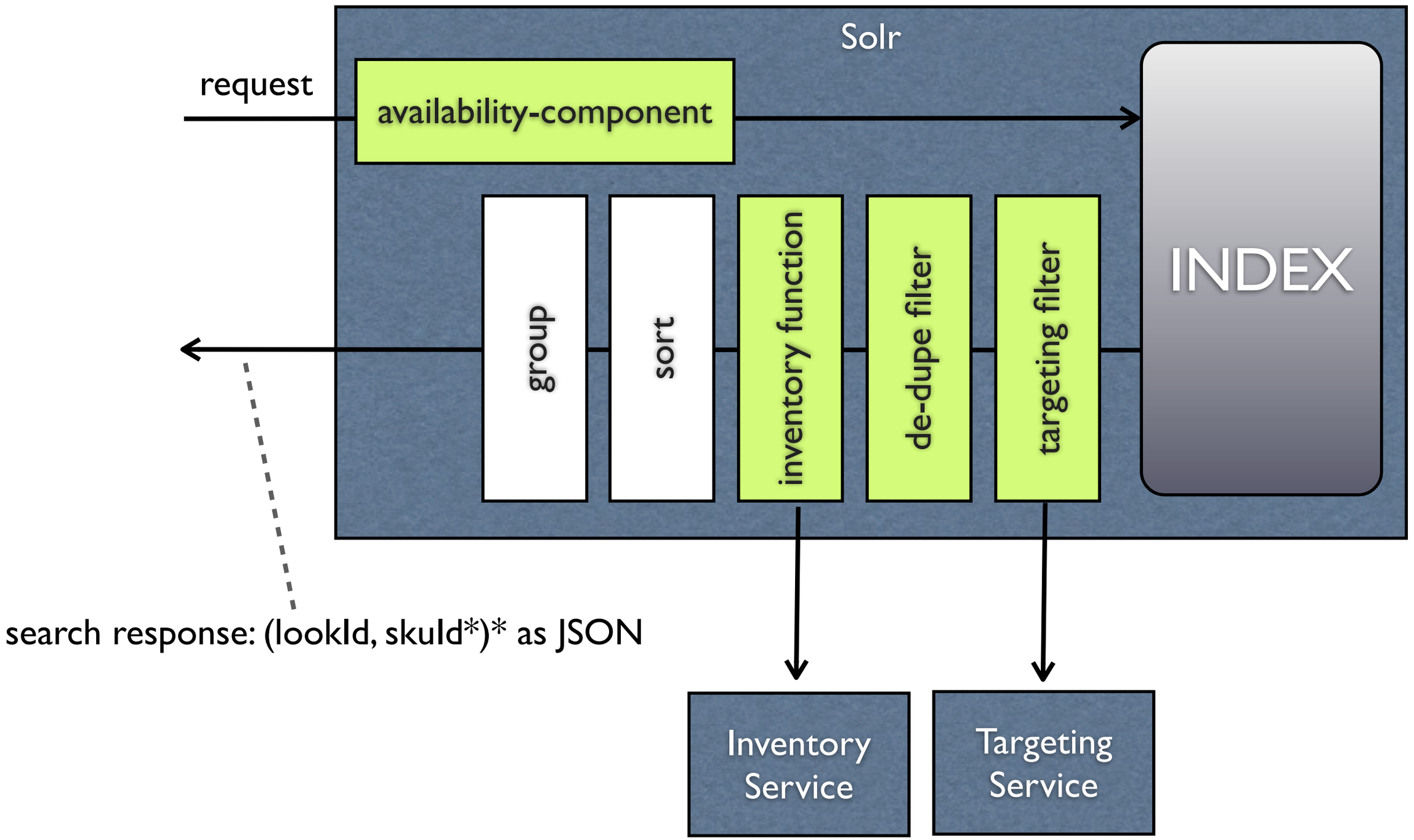
```
sku_name = ...
sku_color = ...
sku_availibility_start =
sku_availibility_end =
sku_store = home
sku_channels = mobile, ipad
sku_sale_id =
```

- The filter becomes much simpler:

```
fq = +sku_availability_start:[* TO NOW/HOUR]
     +sku_availability_end:[NOW/HOUR+1HOUR TO *]
```

- Code fell away. Just had to to post-filter out multiple SKU availabilities in results [O(n)]

Solr Internals & Extensions IV



Faceting

Mixed-logic multi-select faceting

- Facet filters typically use 'AND' logic

```
size = 36 AND brand = 'Timber Island' AND color = 'RED'
```

- Prefer mixed logic, where 'OR' is used for filters on the same facet.

```
size = 36 OR size = 38 AND brand = 'Timber Island'
AND color = 'RED' OR color='BLUE'
```

- Use 'multi-select' faceting technique to ensure that facet values don't disappear.
  - Tag the facet filter-query:
    `fq:{!tag=brand_fq}brand_facet:"Timber Island"`
  - Ignore the effects of the facet filter-query when faceting:
    `<str name="facet.field">{!ex=brand_fq}brand_facet</str>`

‘Disappearing facet problem’

- To get facets relevant to the query, we set facet.mincount to 1.

```
<str name="facet.mincount">1</str>
```

  - However: if subsequent facet filters reduce the facet count to zero, then the facet's 'disappear' :(

- Consider a search for 'shoes', returning the following facets

```
brand: Ade -> 10, Eric -> 5,    color: black -> 8, red -> 7
```

- Then filter on red (and assume that only Ade shoes are red). We want to have:

```
brands: Ade -> 7, Eric -> 0,    colors: black -> 0, red -> 7
```

- However, if `facet.mincount == 1`, we get:

```
brands: Ade -> 7                colors: red -> 7
```

‘Disappearing facet problem’ (cont’)

- We want to say 'Eric and black are relevant to the query, but has zero results due to filtering'.
  - Solution: For each facet overlay the original counts with the values of the new count if present, or zero otherwise.

```
<str name="facet.field">{!ex=brand_fq,size_fq,taxonomy_fq,color_fq
key=all_brand_facets}brand_facet</str>
<str name="facet.field">{!ex=brand_fq,size_fq,taxonomy_fq,color_fq
key=all_color_facets}color_facet</str>
```

- This means we get:

```
all_brand_facets:  Ade -> 10, Eric -> 5,     brand_facet: Ade -> 7
all_color_facets: black -> 8, red -> 7,      color_facet: red -> 7
```

- Merge to get:

```
brand_facet: Ade -> 7, Eric -> 0,  color_facet: red -> 7, black -> 0
```

Color Me Happy

- Need to map SKU's color to a simple set of colors for faceting.
- Used a synonym file to map 'urobilin' to 'yellow'
- Use a stopwords file to remove unknown colors
- It works, but it's brittle; want to move to a solution based on color analysis of swatches

Hierarchical taxonomy faceting

- Our taxonomy is hierarchical. Solr faceting is not :(
- Encoded taxonomy hierarchy using 'paths'

  "Gilt::Men::Clothing::Pants"
- Our search API converts the paths into a tree for rendering purposes.
- Works but (a) feels jenky (b) ordering is alphabetic
- Prefer in future to use unique taxonomy key, and use that the filter through an ordered tree in our AP

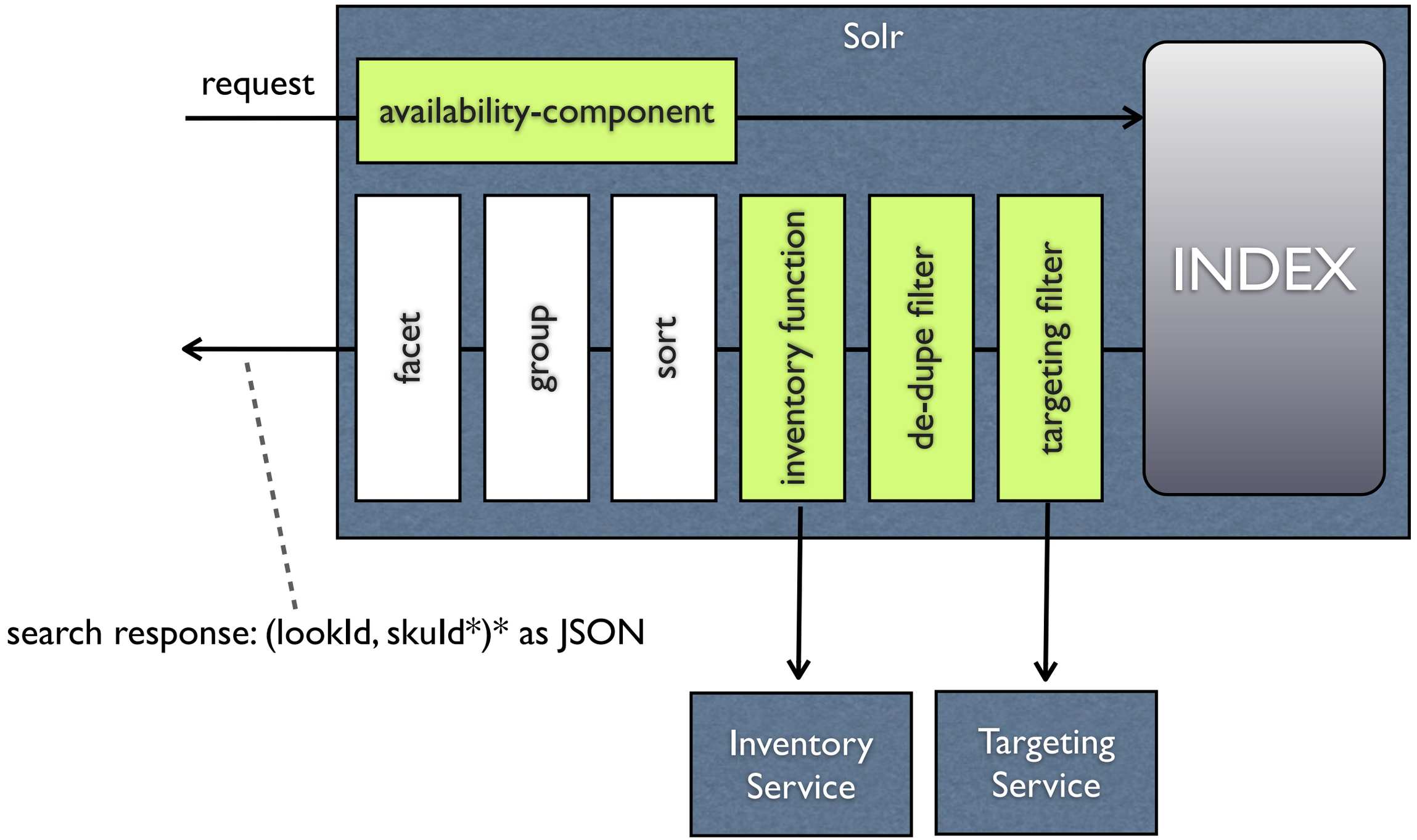
Hierarchical size faceting

- Size ordering is non trivial: there is a two-level hierarchy & ordering is difficult:
  - **00, 0, 2, 4, 6, 8, 10, 12** *rather than* **0, 00, 10, 12, 2, 4, 6, 8**
  - **XS, S, M, L, XL, XXL** *rather than* **M, L, S, XL, XS, XXL**
- End up encoding a size ordinal into the size facet label

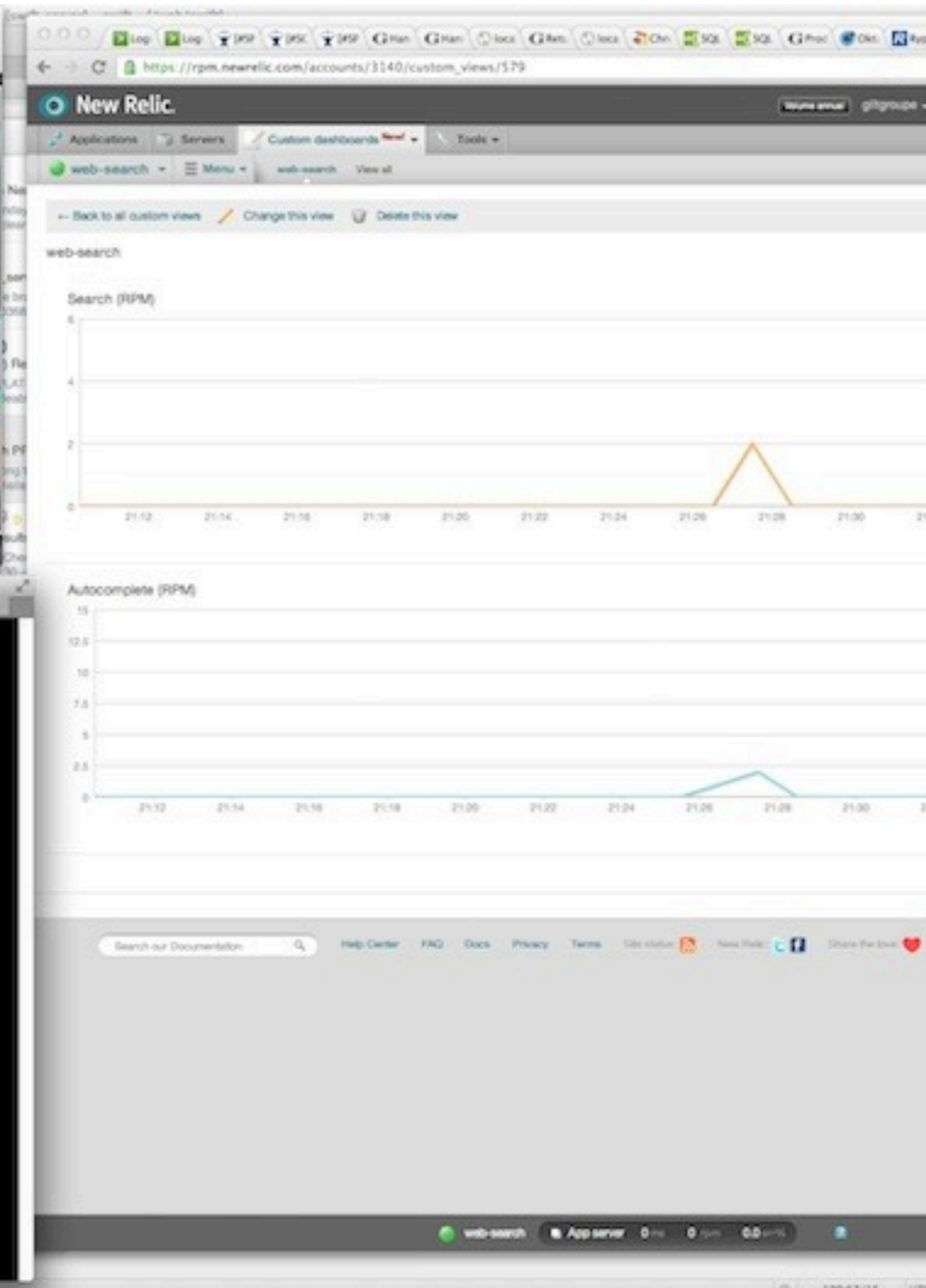
```
"Women's Apparel::[[0000000003]]00"
"Women's Apparel::[[0000000004]]0"
"Women's Apparel::[[0000000005]]2"
```

- OK, but again, feels hacky.
- "If only there was a way to annotate facets with meaningful data"

Solr Internals & Extensions (V)



Lesson I: Solr Makes You Happy

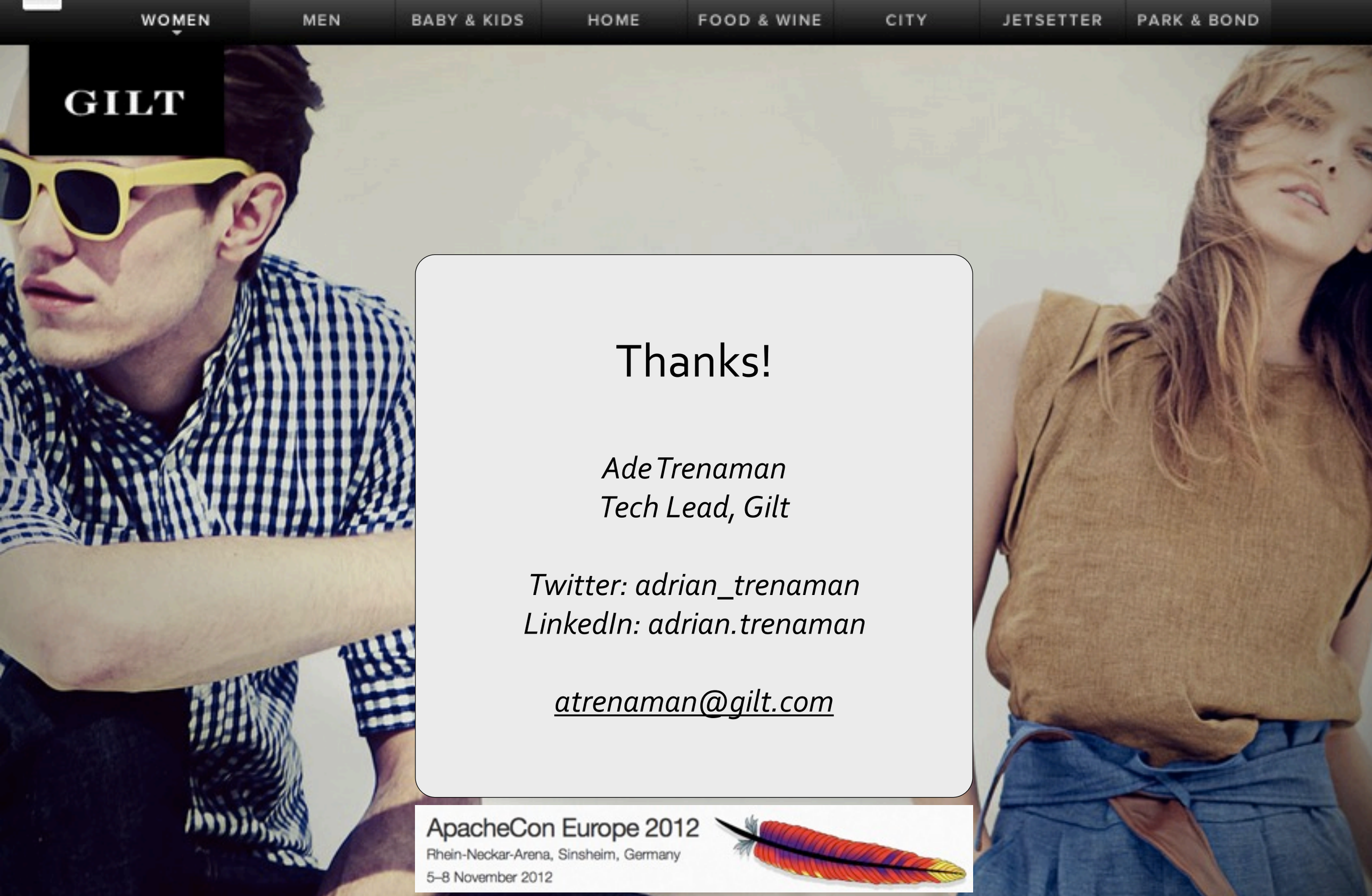


Lesson II: Solr makes the business Love You

- Gilt Search released in A/B test to 30%.
  - Initial deployment can handle **> 180,000 RPM** (5:1 ratio of auto-complete to search)
  - Members who's visit included search are **4x** more likely to buy.
  - Search generates **2-4%** incremental revenue
- Business pleads with us to end A/B test and release to 100%
- We've iterated to drive more traffic (& succeeded) with no loss of conversion.
  - From subtle to suBtle.

Lesson III: Excelsior

- Power *all* listings (keyword, category, sale, brand) via Solr
- Faceting on SKU attributes (e.g. 'age', 'gender')
- Solr 4
- Hack Debridement:
  - Improve hierarchical faceting & facet ordering
  - 'Double facet' feels wrong



Thanks!

Ade Trenaman
Tech Lead, Gilt

Twitter: adrian_trenaman
LinkedIn: adrian.trenaman

atrenaman@gilt.com

ApacheCon Europe 2012
Rhein-Neckar-Arena, Sinsheim, Germany
5–8 November 2012

http://gilt.com/apacheconeu2012

$25

... enjoy!