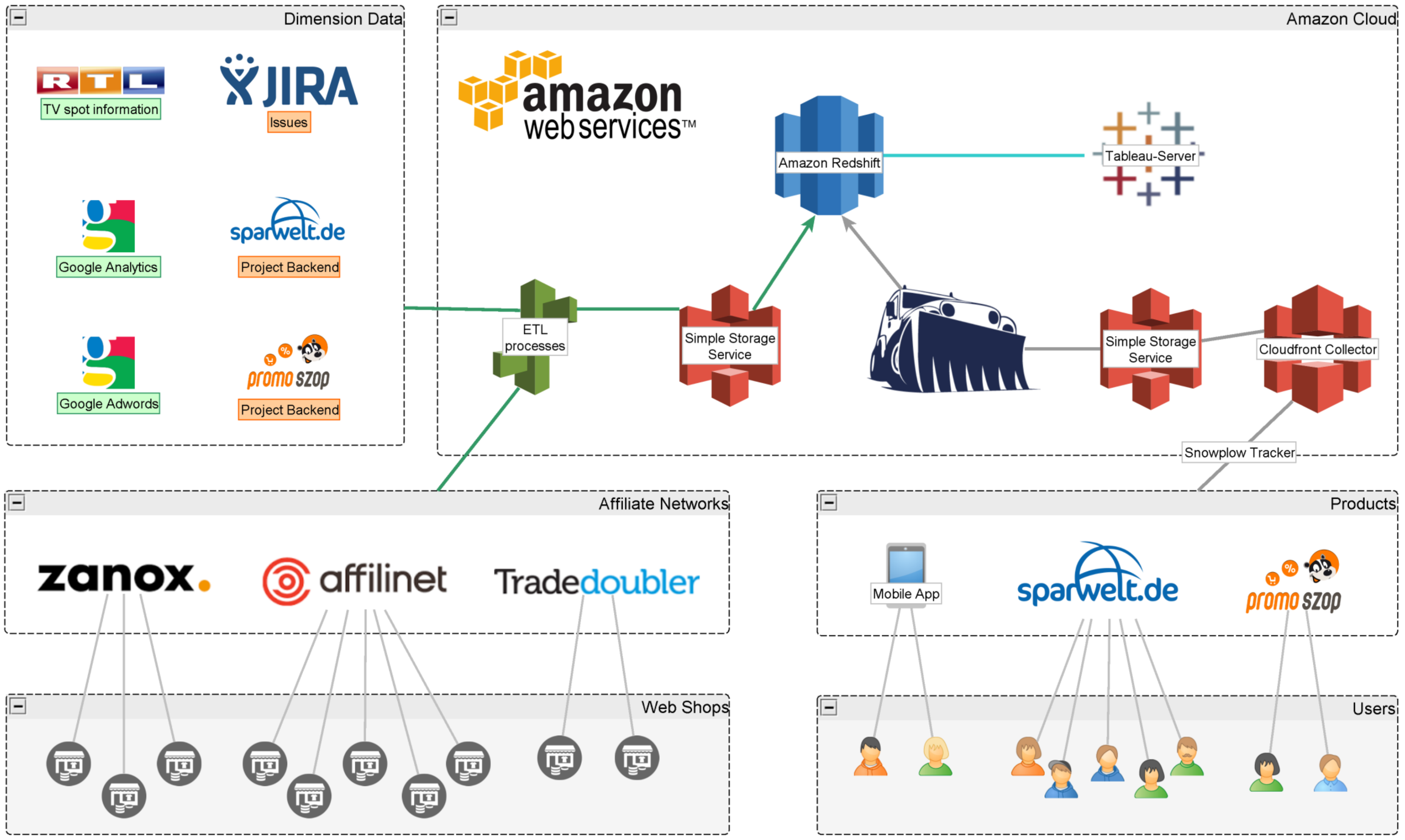
sparwelt.de

Snowplow Meetup 11/08/15

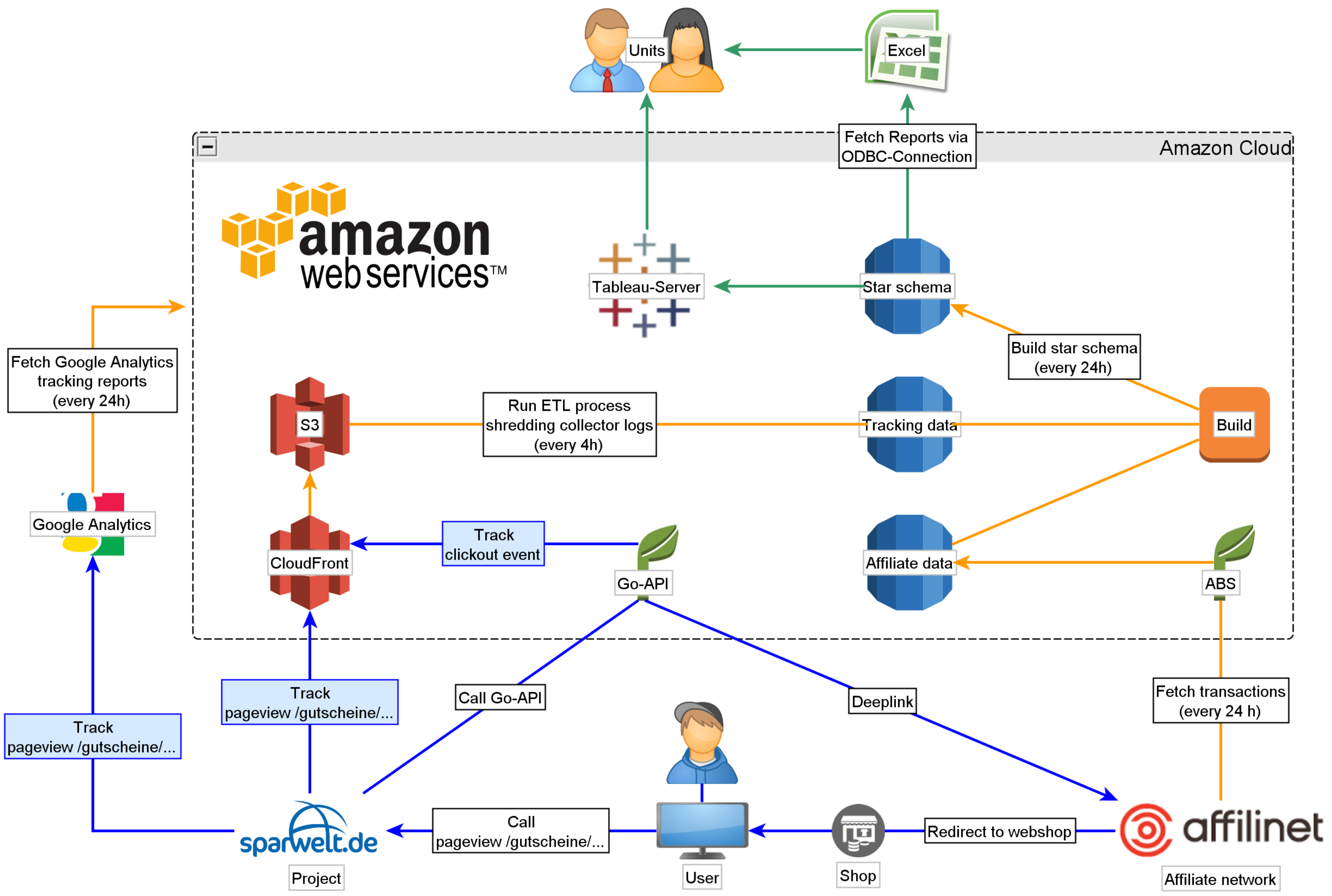
Agenda

1. Introduction
2. System overview
3. Dimensional design
4. Channel attribution
5. General remarks

System overview

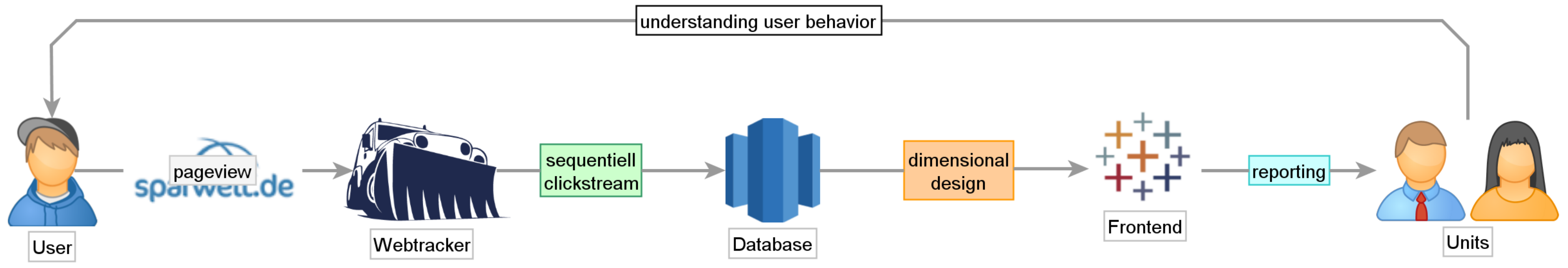


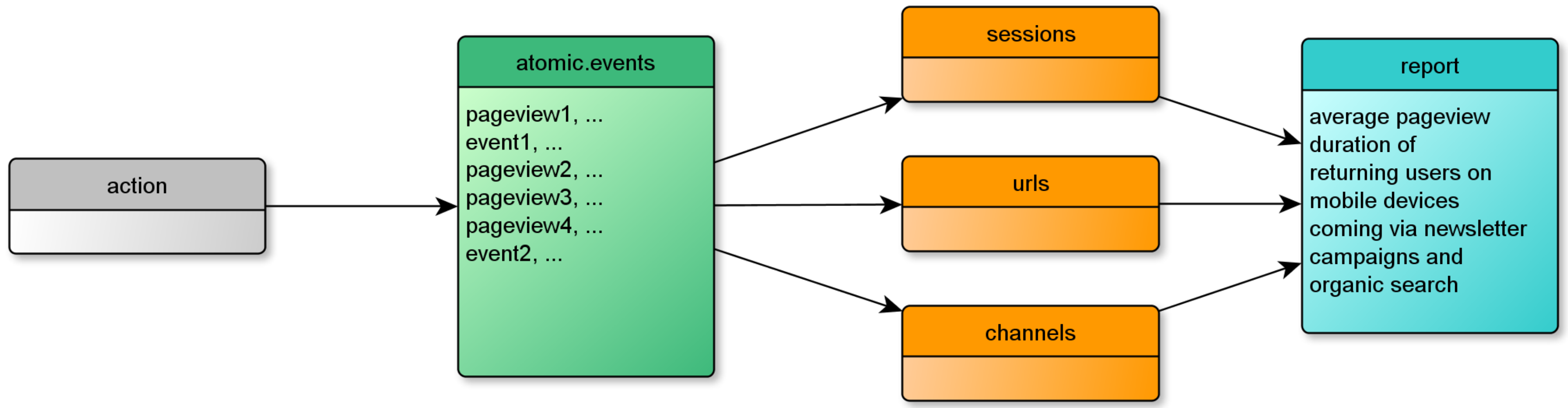
System overview



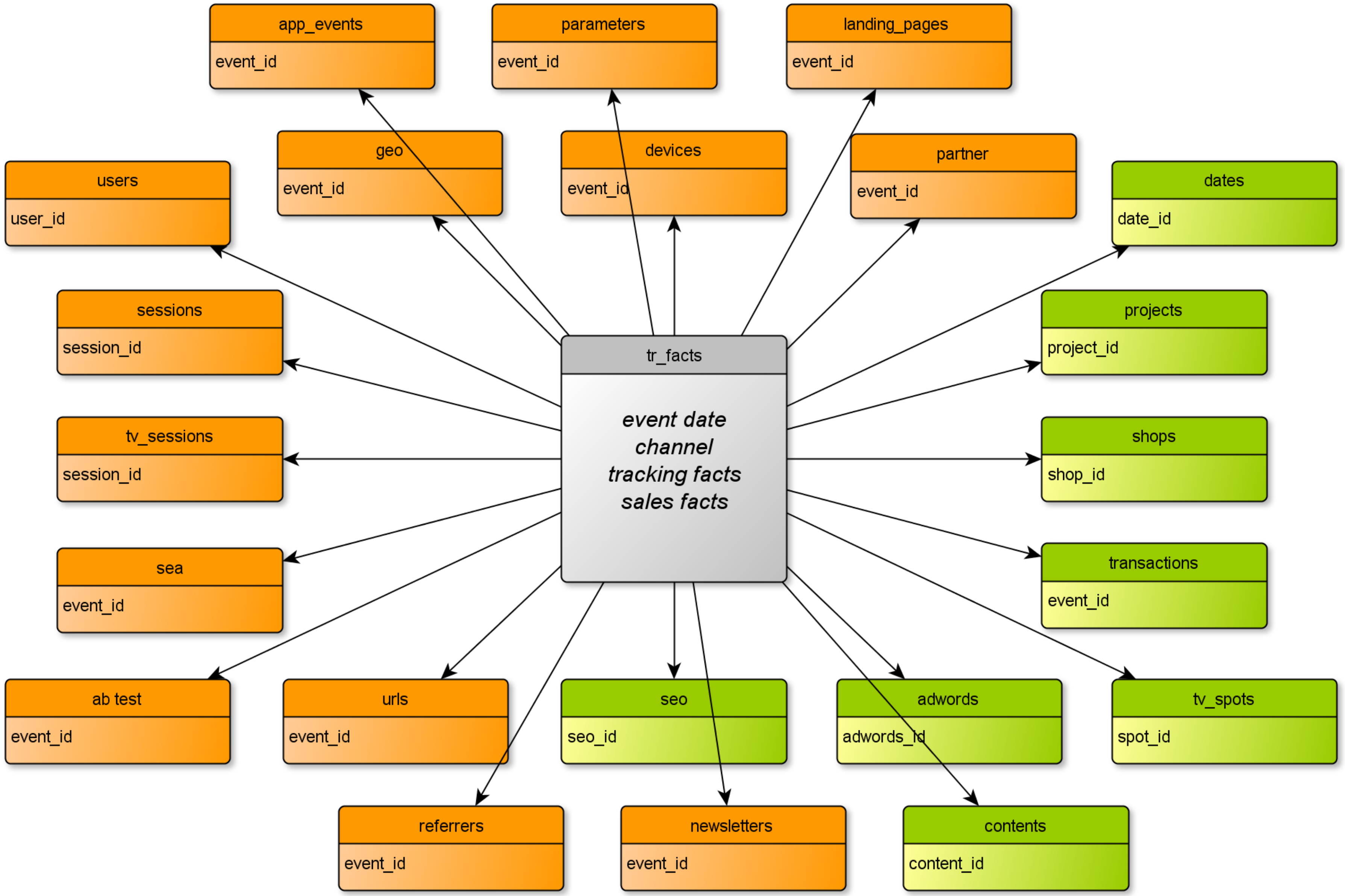
Units
Excel
Fetch Reports via ODBC-Connection
Amazon Cloud
amazon web services™
Tableau-Server
Star schema
Build star schema (every 24h)
Fetch Google Analytics tracking reports (every 24h)
S3
Run ETL process shredding collector logs (every 4h)
Tracking data
Build
Google Analytics
CloudFront
Track clickout event
Go-API
Affiliate data
ABS
Track pageview /gutscheine/...
Call Go-API
Deeplink
Fetch transactions (every 24 h)
Track pageview /gutscheine/...
sparwelt.de
Project
Call pageview /gutscheine/...
User
Shop
Redirect to webshop
affilinet
Affiliate network

Dimensional design





Dimensional design



Performance indicators

- Session Id

```
app_id || '-' || domain_userid || '-' || TO_CHAR(domain_sessionidx,'FM00000') AS session_id
```

- Session metrics

```
RANK() OVER (PARTITION BY session_id ORDER BY event_tstamp ASC, event_id) AS session_hit_idx,
COUNT(*) OVER (PARTITION BY session_id) AS session_hit_count
```

Performance indicators

- Facts for summing

```
SELECT event_id,
       CAST(1 AS SMALLINT) AS pageviews,
       CAST(session_hit_idx = 1 AS SMALLINT) AS sessions,
       CAST(session_hit_idx = 1 AND session_hit_count = 1 AS SMALLINT) AS bounces,
       CAST(session_hit_idx = 1 AND session_idx = 1 AS SMALLINT) AS new_users,
       CAST(session_hit_idx = 1 AND session_idx > 1 AS SMALLINT) AS returning_users,
FROM atomic.events_enriched
WHERE event = 'page_view';
```

Performance indicators

- Facts for distinct counting

```
SELECT event_id,
       CAST(MD5(session_id || page_host || page_path) AS CHAR(32)) AS unique_pageviews,
       domain_userid AS unique_users
FROM atomic.events_enriched
WHERE event = 'page_view';
```

Performance indicators

- Facts for averaging

```
WITH event_metrics AS
(
  SELECT event_id,
         event_tstamp,
         FIRST_VALUE(event_tstamp)
              OVER (PARTITION BY session_id ORDER BY event_tstamp ASC, event_id
                    ROWS BETWEEN 1 FOLLOWING AND 1 FOLLOWING) AS next_event_tstamp
  FROM atomic.events_enriched
  WHERE event = 'page_view'
)
SELECT event_id,
       CAST(DATEDIFF('second',event_tstamp,next_event_tstamp) AS BIGINT) AS pageview_duration
FROM event_metrics;
```

Channel attribution

- Extract URL query parameters

```
SELECT event_id,
       LEFT (NULLIF(REGEXP_SUBSTR (REGEXP_SUBSTR
       (page_query,'utm_source=[^&]*'),'[^=]*$'),''),255) AS utm_source,
       LEFT (NULLIF(REGEXP_SUBSTR (REGEXP_SUBSTR
       (page_query,'utm_campaign=[^&]*'),'[^=]*$'),''),255) AS utm_campaign
       LEFT (NULLIF(REGEXP_SUBSTR (REGEXP_SUBSTR
       (page_query,'vt_network=[^&]*'),'[^=]*$'),''),255) AS vt_network
FROM atomic.events_enriched
WHERE event = 'page_view' AND session_hit_idx = 1;
```

Channel attribution

- Determine referrer medium

```
SELECT event_id,
       CASE
         WHEN refr_medium IS NOT NULL THEN refr_medium
         WHEN refr_host ~ '(facebook|blogger|twitter|gutefrage|blogspot)' THEN 'social'
         WHEN refr_host ~ '(google|yahoo|bing|ask)' THEN 'search'
       END AS refr_medium
FROM atomic.events_enriched
WHERE event = 'page_view' AND session_hit_idx = 1;
```

Channel attribution

```
SELECT event_id AS landing_event_id,
       CASE
         WHEN vt_network = 'g' OR vt_network = 's' THEN 'SEA Search'
         WHEN vt_network = 'd' THEN 'SEA Display'
         WHEN utm_source = 'newsletter' THEN 'Campaign Newsletter'
         WHEN utm_source = 'facebook' THEN 'Campaign Facebook'
         WHEN utm_source IS NOT NULL THEN 'Campaign Other'
         WHEN refr_medium = 'social' THEN 'Social'
         WHEN refr_medium = 'search' AND page_path = '/' THEN 'Brand Search'
         WHEN refr_medium = 'search' AND LEFT(lr.refr_path,7) = '/imgres' THEN 'Image Search'
         WHEN refr_medium = 'search' THEN 'Organic Search'
         WHEN (refr_url IS NULL AND session_referrers = 0 AND session_pageviews > 1)
              OR refr_url = 'blockedReferrer' THEN 'Blocked Referrer'
         WHEN (session_referrers > 0 OR (session_referrers = 0 AND session_pageviews = 1))
              AND refr_url IS NULL THEN 'Direct'
         WHEN page_host != refr_host THEN 'Referrer'
         WHEN page_host = refr_host THEN 'Session Timeout'
       END AS channel
FROM atomic.events_enriched
WHERE event = 'page_view' AND session_hit_idx = 1;
```

Channel attribution

```
WITH event_metrics AS (
  SELECT domain_userid, event_id, event_tstamp, refr_domain,
          LAST_VALUE(CASE event WHEN 'page_view' THEN page_domain END IGNORE NULLS)
               OVER(PARTITION BY user_id
                    ORDER BY event_tstamp ASC,event DESC, tr_events.event_id
                    ROWS BETWEEN UNBOUNDED PRECEDING AND 1 PRECEDING
                    ) AS previous_page_domain
  FROM atomic.events_enriched
  WHERE event = 'page_view'
)
SELECT event_id,
       LAST_VALUE(CASE WHEN (refr_domain != NVL(previous_page_domain,'') OR refr_domain IS NULL)
                           AND session_hit_idx = 1 THEN event_id
                  END IGNORE NULLS)
            OVER(PARTITION BY domain_userid
                 ORDER BY event_tstamp ASC, event_id
                 ROWS UNBOUNDED PRECEDING
                ) AS landing_event_id
FROM event_metrics;
```

General recommandations

1. Prepare all dimensions with unique primary keys
2. Expand large dimensions to map the dimension keys to the `event_id` of the corresponding events
3. Create a hub table holding all dimension keys with the `event_id` being your distribution key
4. Aggregate measures to event, session or date level in separate measure tables
5. Create different fact tables based on the hub table for reporting