PRIVACY @INSTAGRAM

Carrie Jiang — 2018

PRIVACY @INSTAGRAM

Agenda

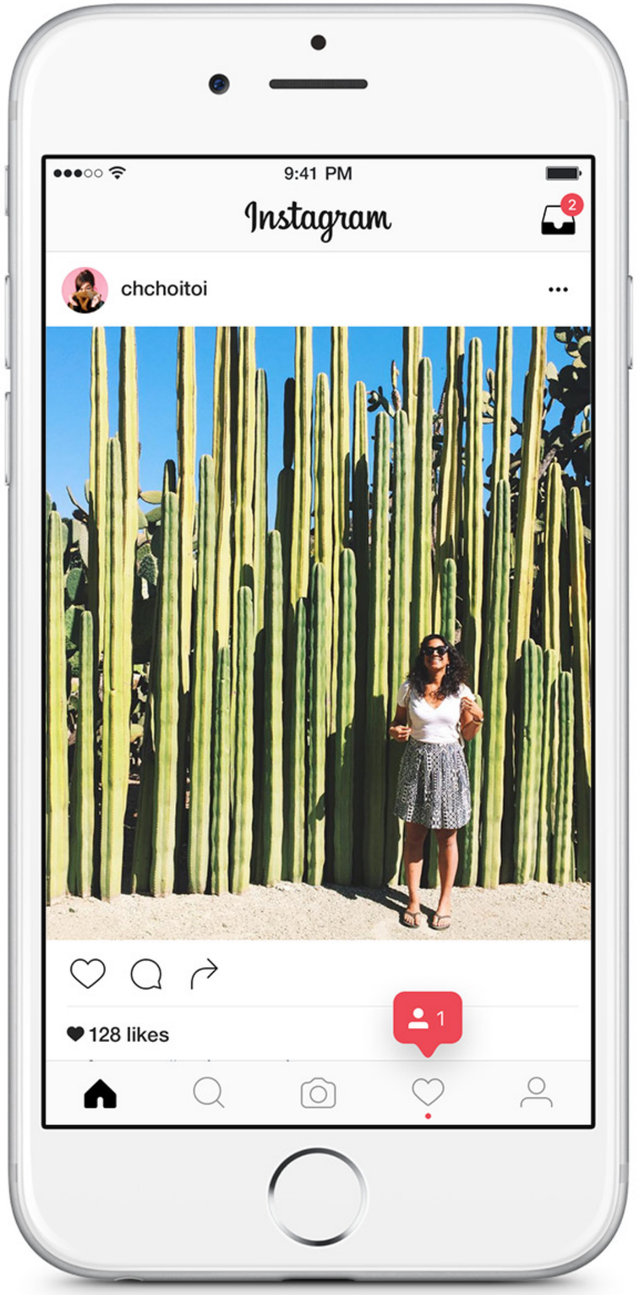
1 **The initiative**

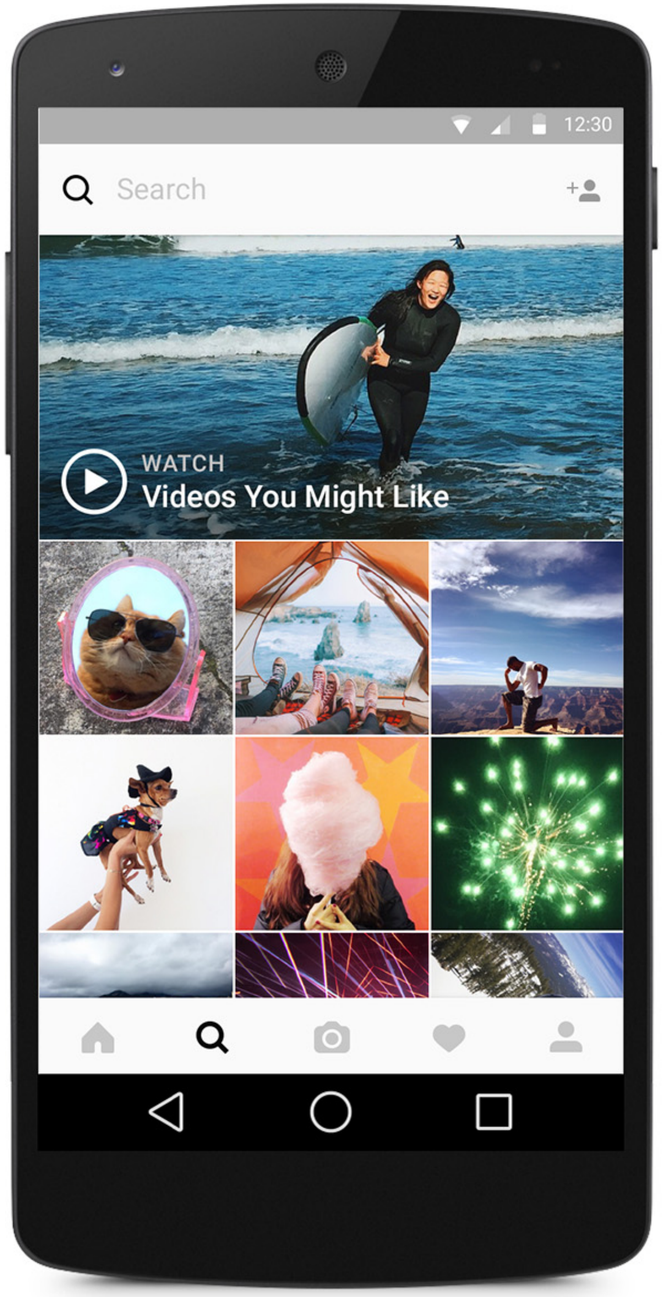
---

2 **The framework**

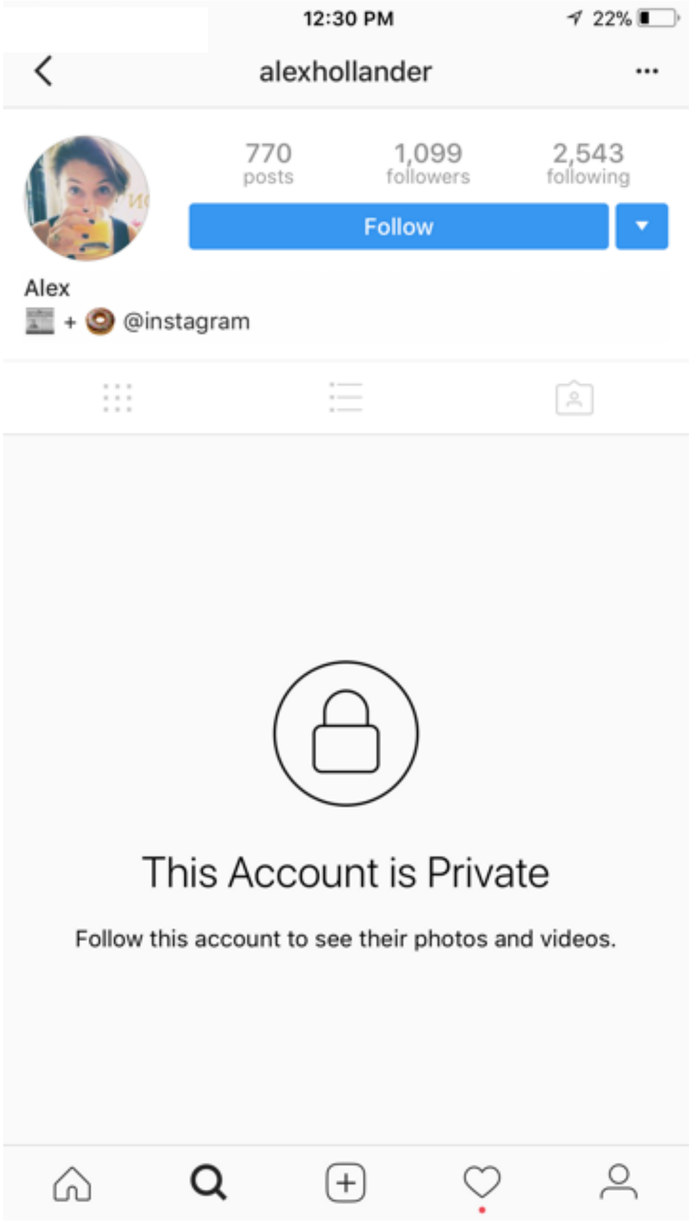
---

3 **Privacy policy inference**


9:41 PM
Instagram
chchoitoi
128 likes


12:30
Search
WATCH
Videos You Might Like


8:08 PM
48%
instagram
5,009
posts
230M
followers
197
following
Message
Instagram
Discovering — and telling — stories from around the world.
Curated by Instagram's community team.
blog.instagram.com/
Followed by wake.to.sleep, cyscottchen, fdeliege + 28 more


12:30 PM
22%
alexhollander
770
posts
1,099
followers
2,543
following
Follow
Alex
+ @instagram
This Account is Private
Follow this account to see their photos and videos.

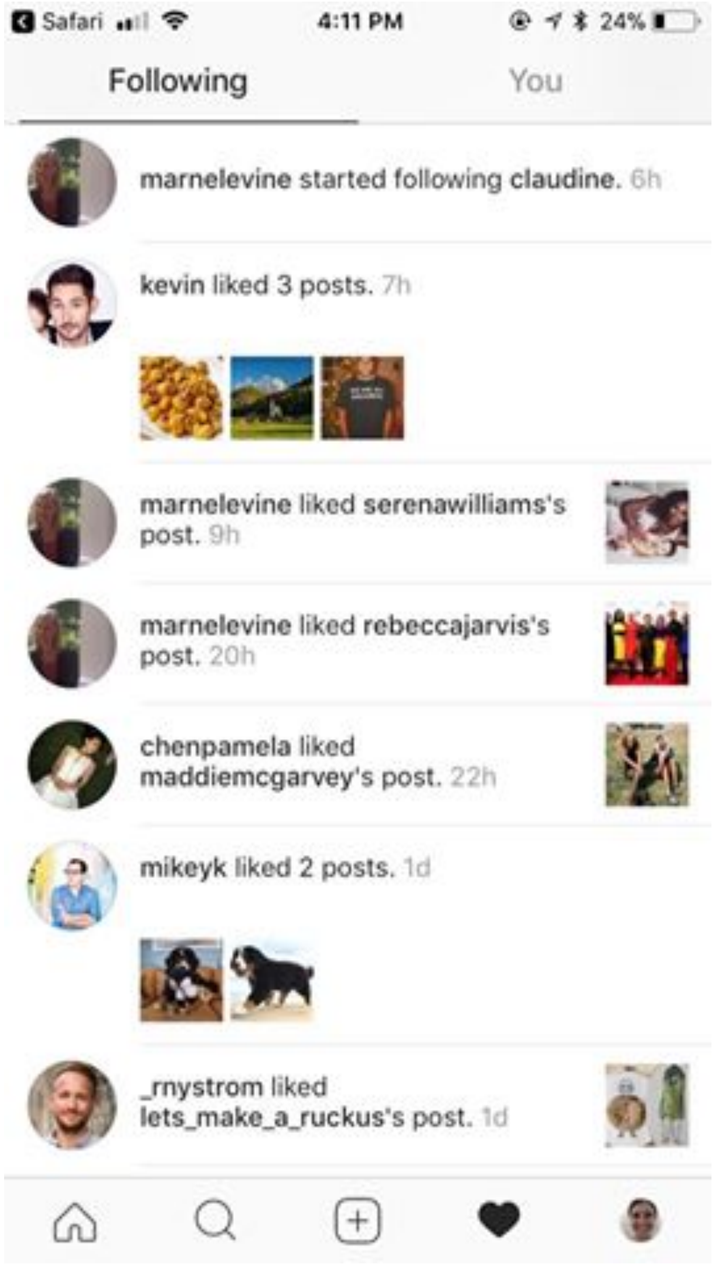
INSTAGRAM USER GROWTH



#hashtags


Insta


DISCOVER
NEW PEOPLE
EXPLORE
TRENDING PLACES
Insta
GLIMMERING
ISLANDS
EXTREME
ATHLETES

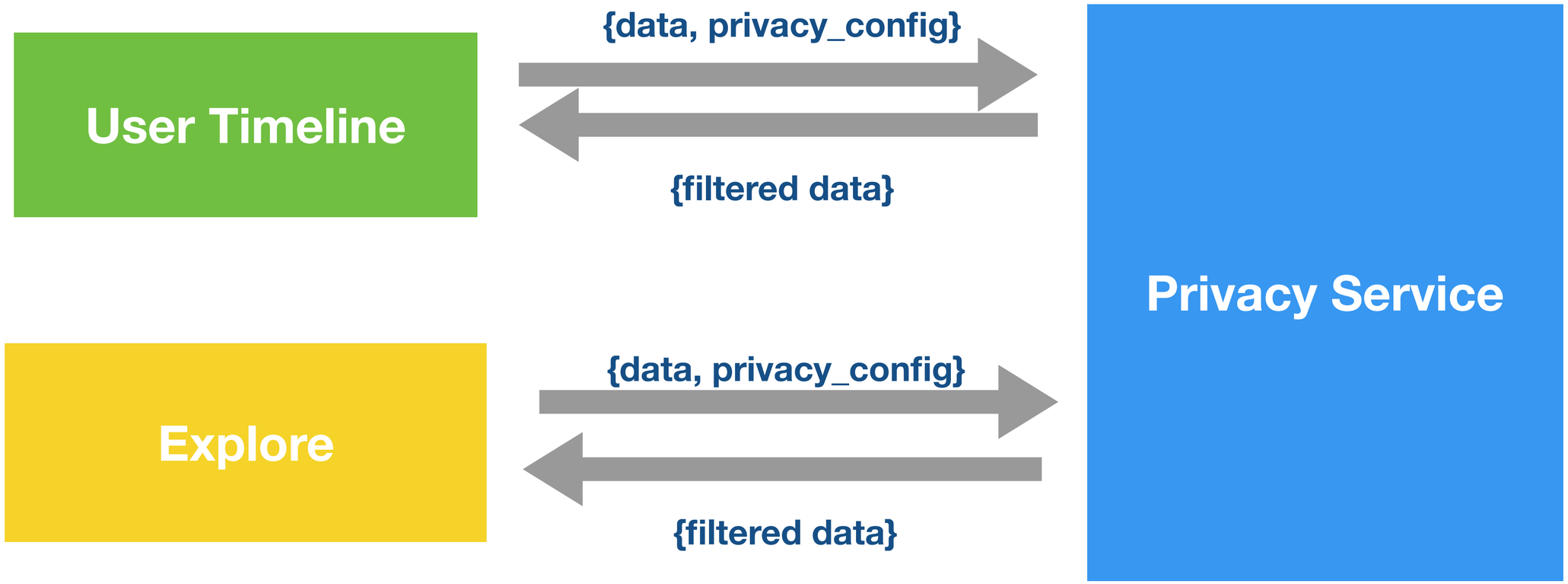

Safari
4:11 PM
24%
Following
You
marnelevine started following claudine. 6h
kevin liked 3 posts. 7h
marnelevine liked serenawilliams's post. 9h
marnelevine liked rebeccajarvis's post. 20h
chenpamela liked maddiemcgarvey's post. 22h
mikeyk liked 2 posts. 1d
_rnystrom liked lets_make_a_ruckus's post. 1d

```
def filter_activities(activities, viewer):
    result = []
    for act in activities:
        if is_blocking(viewer, act.actioner) or blocked(viewer, act.actioner):
            continue
        if act.type == LIKE:
            if is_blocking(viewer, act.target_user) or \
                    blocked(viewer, act.target_user):
                continue
            if act.target_user.is_private and \
                    not is_following(viewer, act.target_user):
                continue
            if not can_view_media(viewer, act.media):
                continue
            result.append(act)
        elif act.type == FOLLOW:
    return result
```

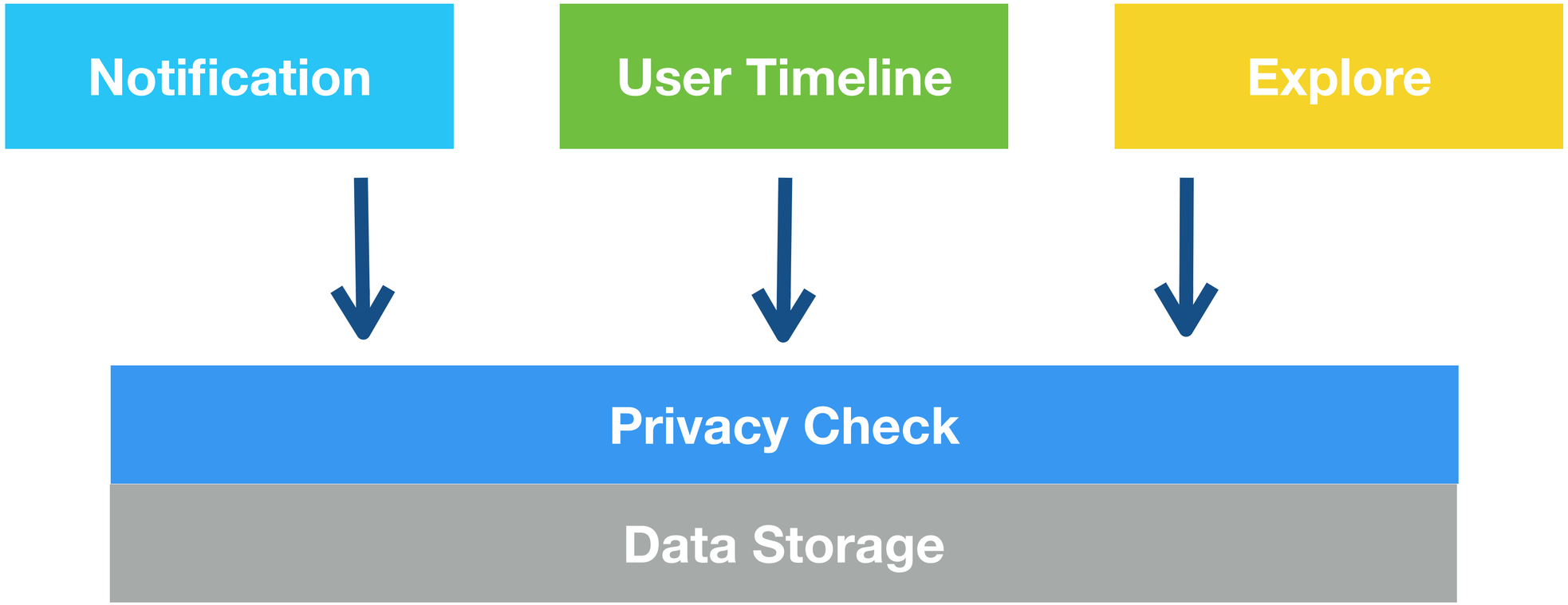
CENTRALIZED PRIVACY SERVICE

- **Endpoint level privacy check**
- **Data level privacy check**

ENDPOINT LEVEL PRIVACY



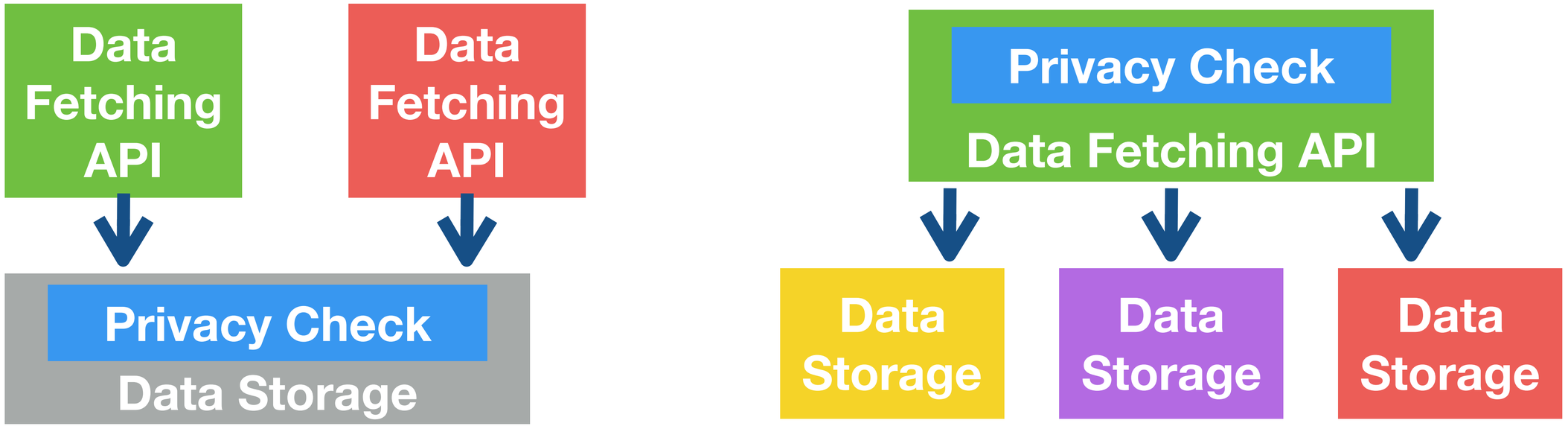
DATA LEVEL PRIVACY



COMPARISON

	Correctness	Performance	Developer Velocity
Endpoint Level	Flexible per use case High maintenance	Overfetch	Master privacy settings
Data Level	Enforced everywhere Low maintenance	Proper caching	Implicit and automatic

DATA LEVEL PRIVACY



TERMINOLOGY

- **Viewer Context**

- **Privacy Rule**

  - allow or deny decision on privacy scenario

- **Privacy Policy**

  - allow or deny decision on an object based on a series of privacy scenarios

PRIVACY RULE

```
class DenyIfUserIsBlockedRule:
    DENY_RULING = Ruling.deny(
        'DenyIfUserIsBlockedRule',
        'Deny if target user and viewer are blocking each other'
    )

    def evaluate(
        self,
        viewer_context: ViewerContext,
        node: "NodeUser",
    ) -> Ruling:
        if node.blocked(viewer_context.user) or node.is_blocking(viewer_context.user):
            return DENY_RULING
        return skip_ruling
```

PRIVACY POLICY ON USER

```
read_policy:
  -deny_if_user_is_blocked_rule
  -deny_if_user_is_gated_rule
  -allow_if_public_user_rule
  -always_deny_rule
```

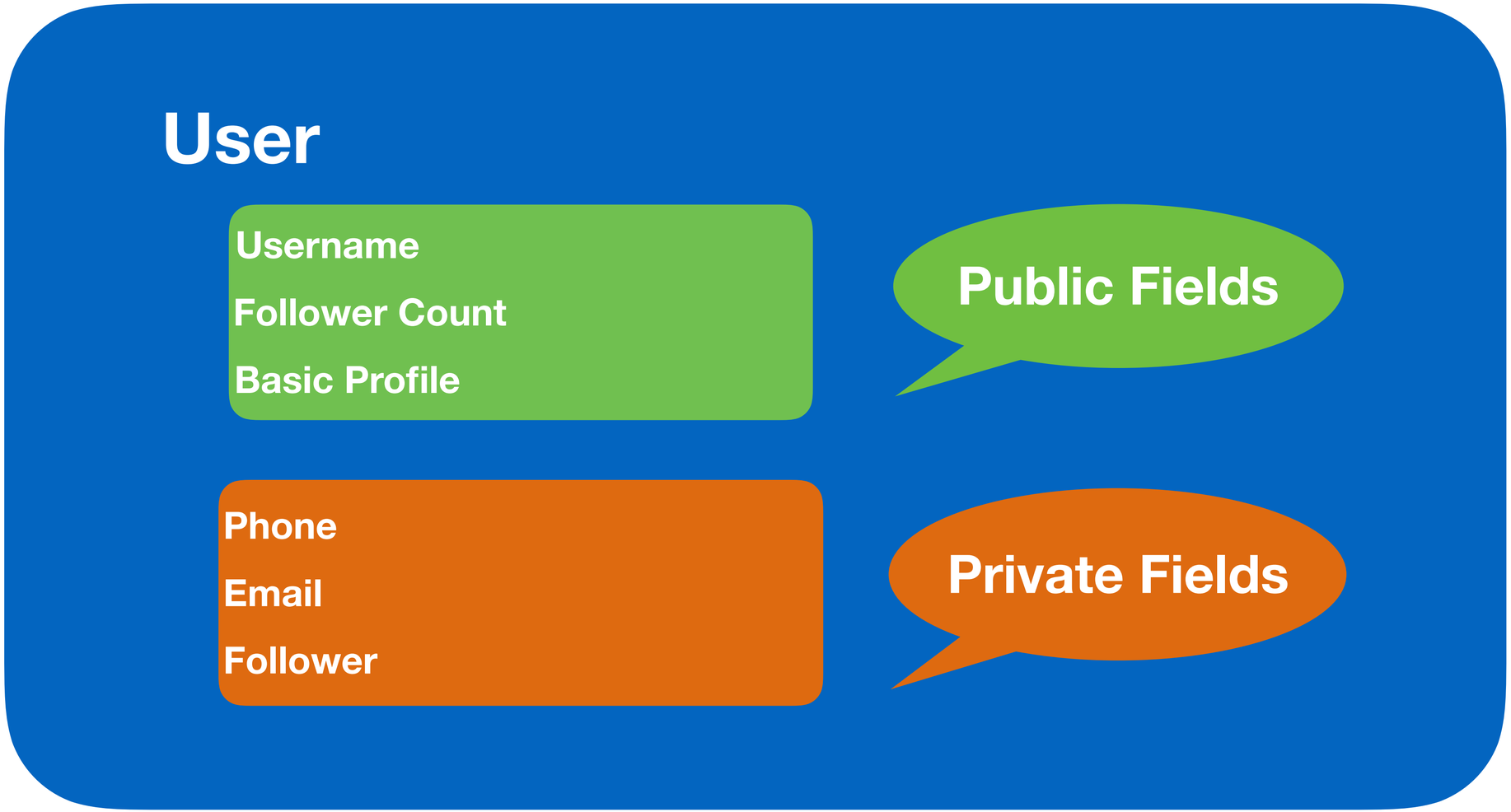
DELEGATION

```
read_policy:
  -deny_if_cannot_see_media_owner_rule
  -allow_if_media_owner_is_public_rule
  -always_deny_rule
```

EDGE PRIVACY



FIELD PRIVACY



DEPLOYMENT

- **Focus on core types that responsible for most of the data access**
- **Always stricter or equal to existing privacy checks**
- **REALLY Look out for performance**

POLICY INFERENCE

- **Sample data accesses per endpoint**
- **Build relationship graph between viewer and data**
- **Analyze common relationship**

POLICY INFERENCE

Device ID	Viewer	Owner	Endpoint
1	Joe	Joe	user.settings
2	Alex	Alex	explore
3	Mike	Mike	user.login
4	Emily	Emily	user.settings
5	None	Kevin	user.foo

CONCLUSION

- Data level privacy check
  - correctness
  - develop velocity
- Privacy policy inference
  - infer policy
  - detection



