



RESTful Data Services with LinkRest

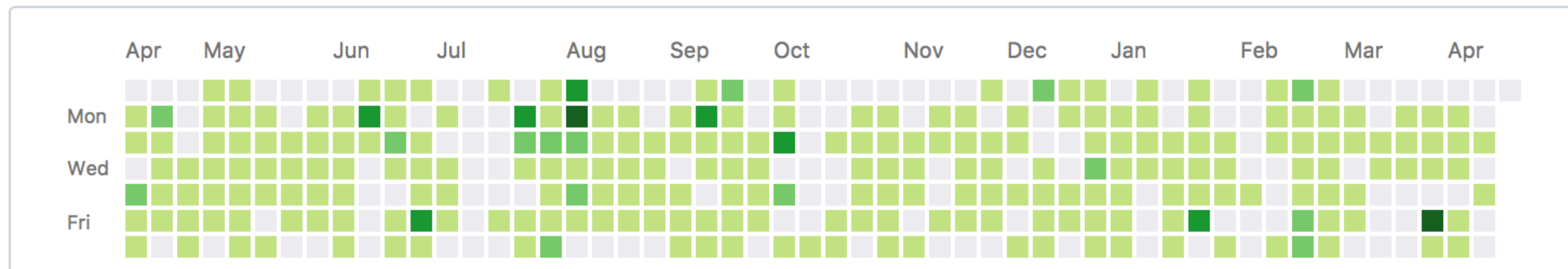
by Andrus Adamchik
@andrus_a

About Me

- Open source developer: ORM, DI, REST, no-container
- Member and ex-VP of the Apache Software Foundation
- Run a company called ObjectStyle

2,568 contributions in the last year

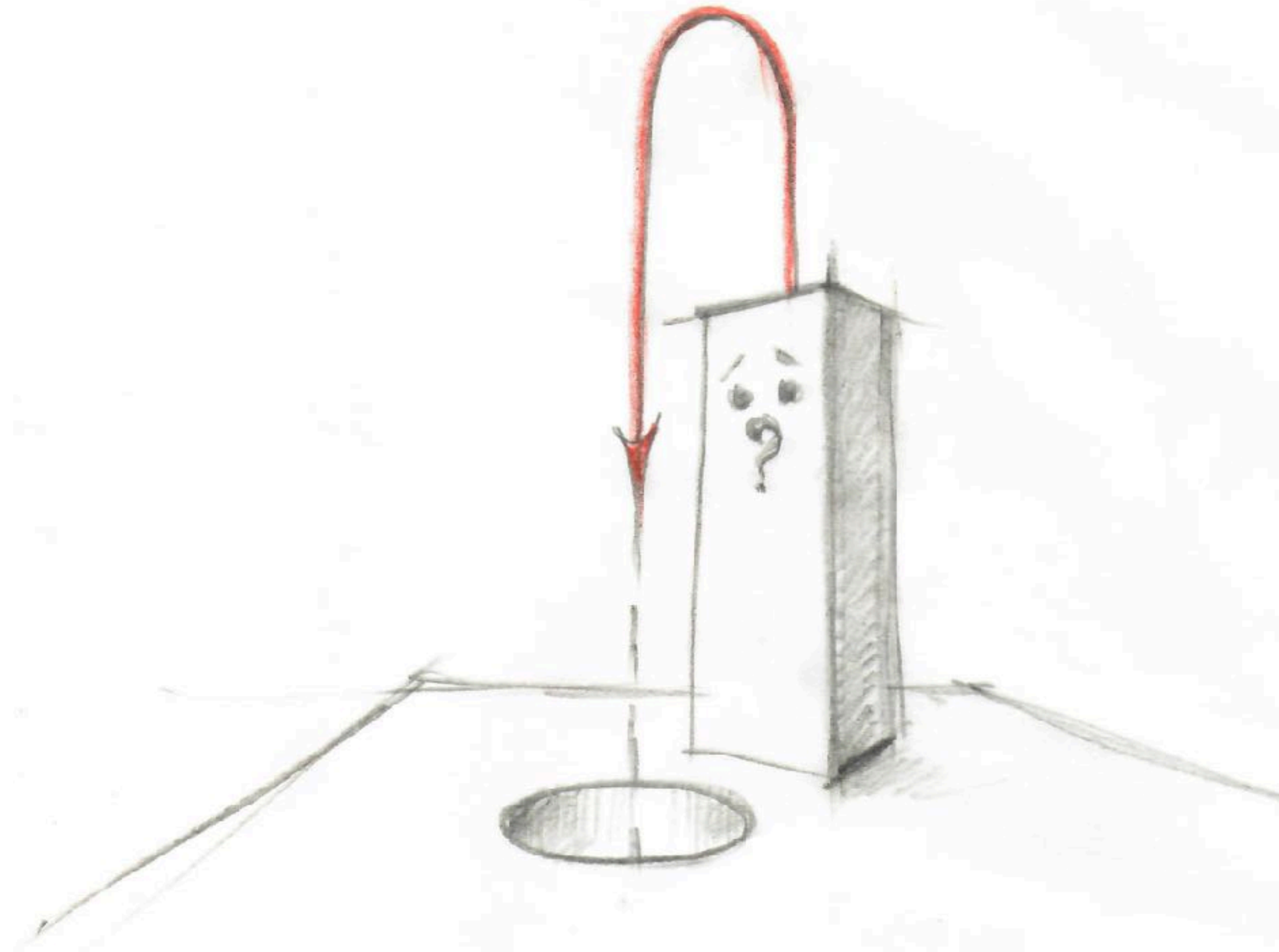
Contribution settings ▾



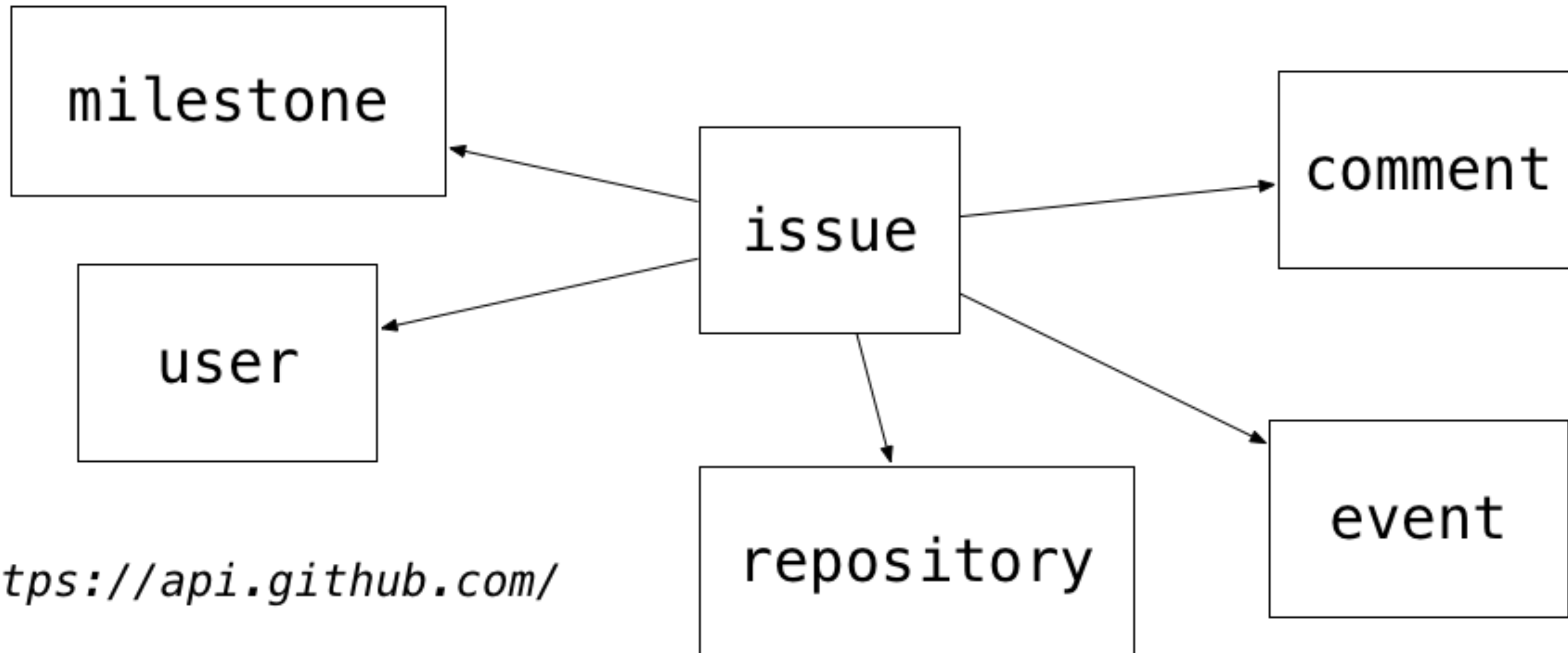
I Love REST

- HTTP is a clean and universal protocol
- Near-infinite scalability

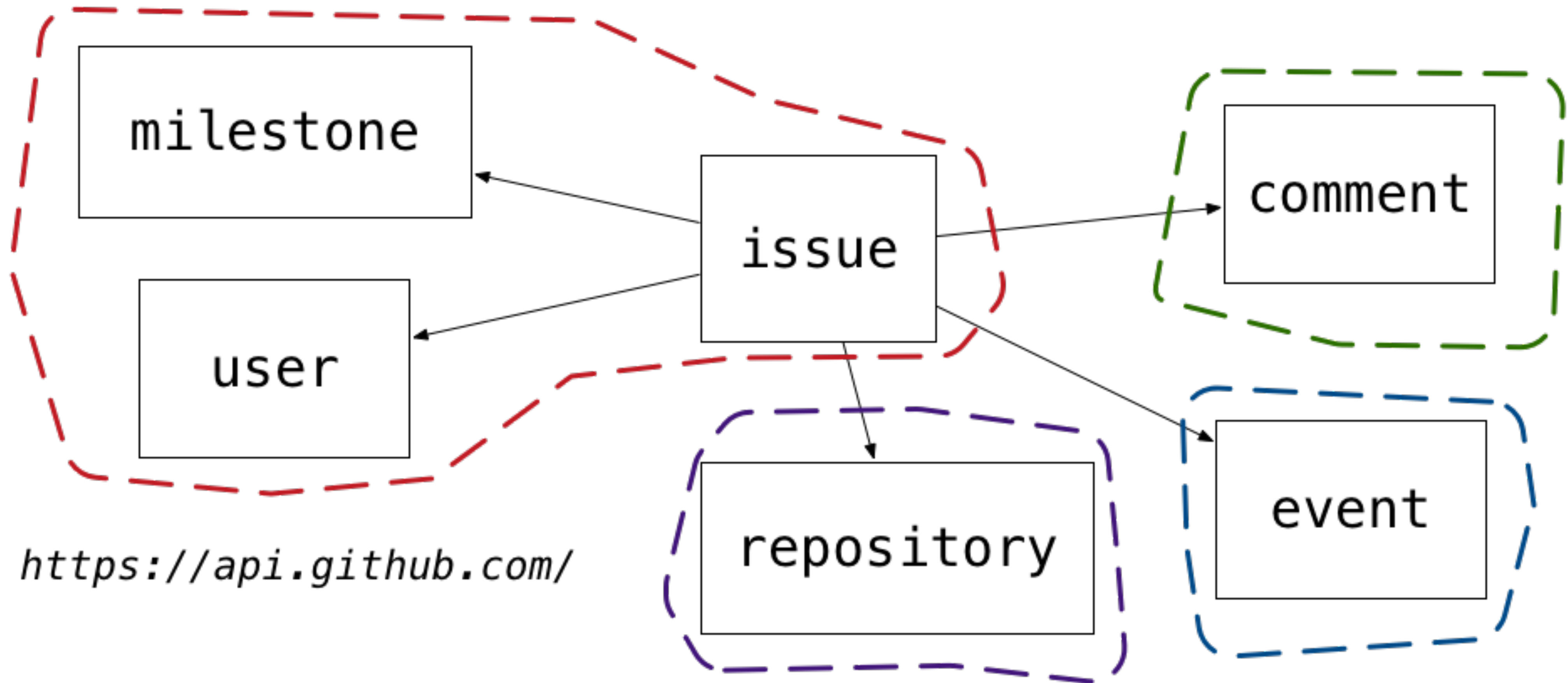
Hi! I am a REST API Client



Domain Model



Domain Model Aggregates



Aggregate-Centric REST API

- An aggregate handled by a single REST resource
- Decoupling / encapsulation
- Assumes "normal" use cases don't cross aggregate boundaries

Problem #1: Too Much Data

```
curl -i \  
  'https://api.github.com/repos/nhl/link-rest/issues?state=all'
```

- Returns **126K** JSON
- We need **3K** ("title", "number", "id")

Problem #2: Too Little Data

```
curl -i \  
  'https://api.github.com/repos/nhl/link-rest/issues?state=all'
```


















```
{  
  // O(N) problem:  
  // for each issue fetch events via a separate REST call  
  "events_url" : "https://api.github.com/.../issues/207/events"  
  ...  
}
```

LinkRest



- From aggregates to graphs
- Client decides on request shape / size
- ~~One size fits all~~
- Reuse existing (ORM) models

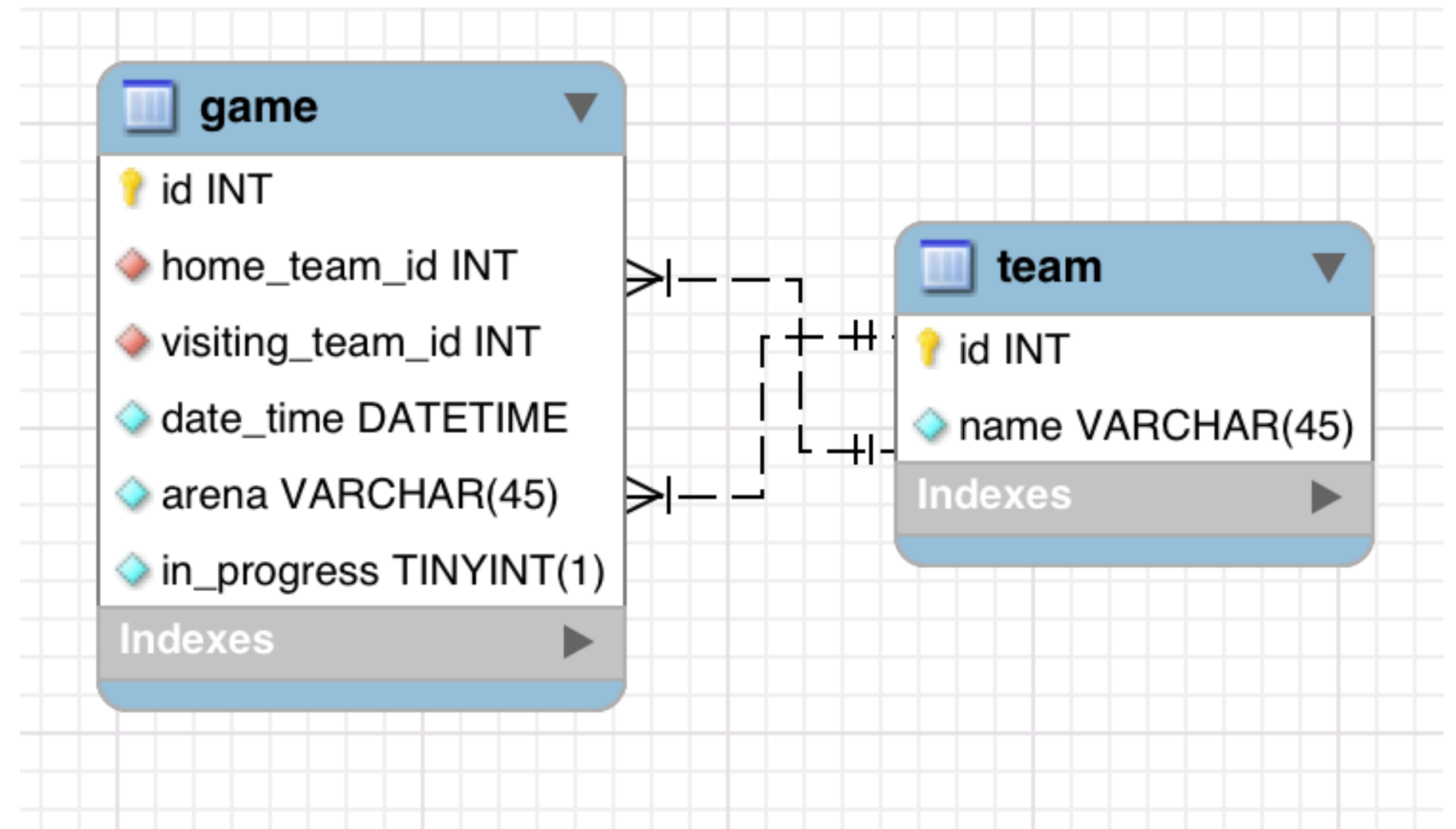
Demo: Hockey Schedule App

Sunday, Oct 30

Matchup	Time	Networks
 Buffalo @  Winnipeg	3:00 PM ET	TSN3, MSG-B
 Philadelphia @  Carolina	5:00 PM ET	FS-CR, TCN-PH
 Florida @  Detroit	5:00 PM ET	FS-D+, FS-F
 Toronto @  NY Islanders	6:00 PM ET	SN360, SN1, MSG+
 Los Angeles @  Chicago	7:00 PM ET	CSN-CH, FS-W
 Tampa Bay @  NY Rangers	7:00 PM ET	 TVAS, MSG, SUN
 Ottawa @  Edmonton	9:00 PM ET	SNOL, RDSI, TSN5
 Washington @  Calgary	9:30 PM ET	SNF, CSN-DC+

Tuesday, Nov 1

Matchup	Time	Networks
 St. Louis @  NY Rangers	7:00 PM ET	MSG+, FS-MW



LinkRest Protocol

Shaping Response

```
?include=id
```

```
{"id":2016020001}
```

```
?include=id&include=homeTeam.name
```

```
{"id":2016020001,"homeTeam":{"name":"Rangers"}}
```

```
?exclude=arena&exclude=datetime
```

```
{"id":2016020001,"inProgress":false}
```

Filtering

```
?cayenneExp=inProgress=true
```

```
SELECT t0.* FROM game t0 WHERE t0.in_progress = 1
```

```
?cayenneExp=["homeTeam.name like $t","New York%"]
```

```
SELECT t0.* FROM game t0 JOIN team t1 ON (t0.home_team_id =  
t1.id) WHERE t1.name LIKE 'New York%'
```

Sorting

```
?sort=dateTime
```

```
SELECT t0.* FROM game t0 ORDER BY t0.date_time
```

```
?sort=dateTime&dir=DESC
```

```
SELECT t0.* FROM game t0 ORDER BY t0.date_time DESC
```

```
?sort=[{"property":"inProgress"},  
{"property":"homeTeam.name","direction":"DESC"}]
```

```
SELECT t0.* FROM game t0 JOIN team t1 ON (t0.home_team_id =  
t1.id) ORDER BY t0.in_progress, t1.name DESC
```

Pagination *(for a list of 10 objects)*

```
?start=2
```

```
{"data": [ /*objects 2..9*/ ], "total":10}
```

```
?limit=5
```

```
{"data": [ /*objects 0..4*/ ], "total":10}
```

```
?start=2&limit=5
```

```
{"data": [ /*objects 2..6*/ ], "total":10}
```


Advanced Include

```
?include={"path":"homeGames","cayenneExp":"arena = 'Madison Square Garden'", "sort":"dateTime"}&include=homeGames.id
```

```
{  
  "homeGames": [  
    { "id": 2016020075 },  
    { "id": 2016020790 }  
  ]  
}
```

Constraints API


















```
ConstraintsBuilder<Team> tc = Constraint.idAndAttributes(Team.class);
ConstraintsBuilder<Game> gc = Constraint.idAndAttributes(Game.class)
    .path("homeTeam", tc)
    .path("visitingTeam", tc);

return LinkRest.select(Game.class, configuration)
    .uri(uri)
    .constraint(gc)
    .get();
```



- Prevents DOS attacks
- Fine-grained attribute security

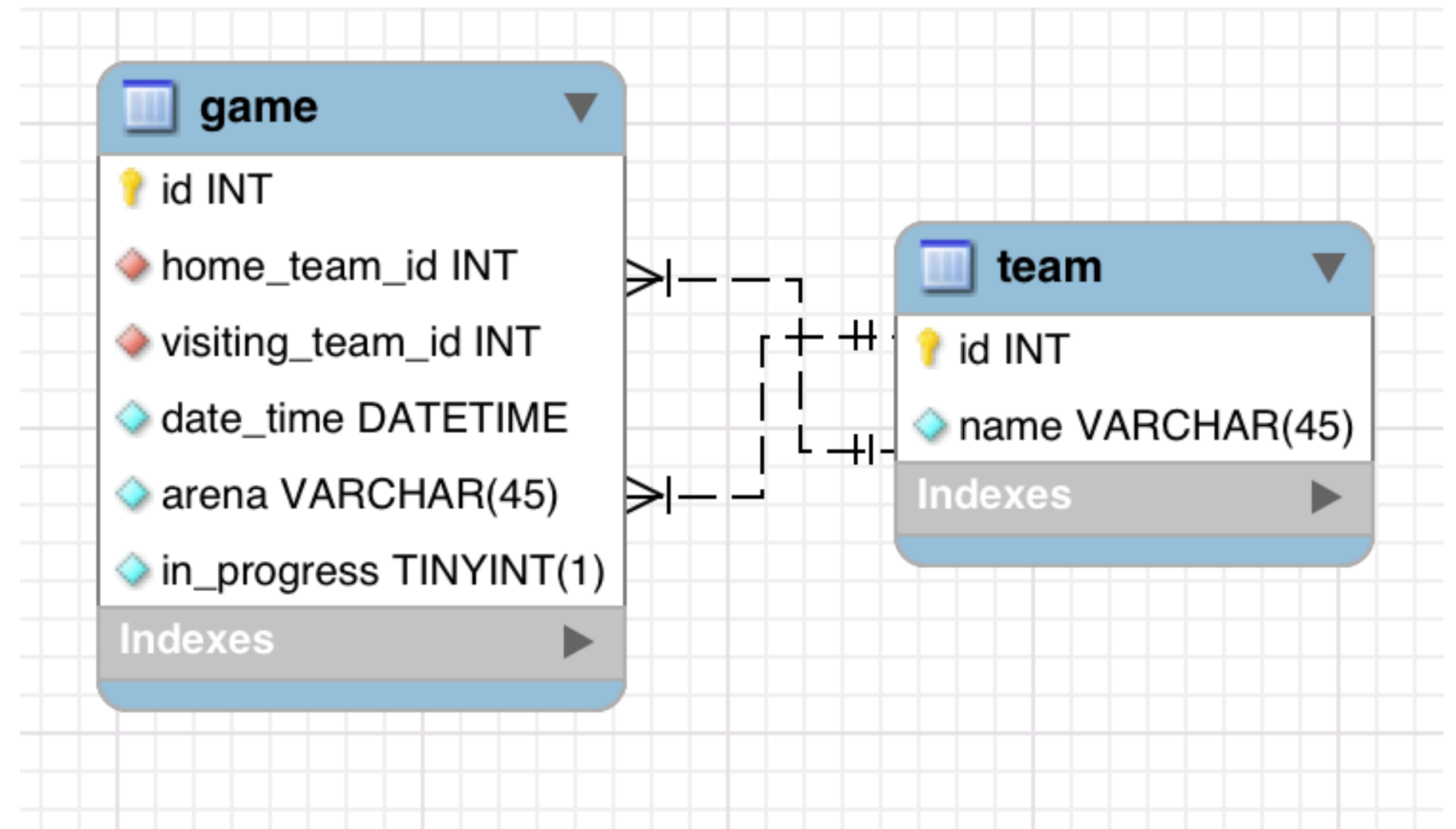
Demo: Modifying Data

Sunday, Oct 30

Matchup	Time	Networks
 Buffalo @  Winnipeg 3:00 PM ET TSN3, MSG-B		
 Philadelphia @  Carolina 5:00 PM ET FS-CR, TCN-PH		
 Florida @  Detroit 5:00 PM ET FS-D+, FS-F		
 Toronto @  NY Islanders 6:00 PM ET SN360, SN1, MSG+		
 Los Angeles @  Chicago 7:00 PM ET CSN-CH, FS-W		
 Tampa Bay @  NY Rangers 7:00 PM ET  TVAS, MSG, SUN		
 Ottawa @  Edmonton 9:00 PM ET SNOL, RDSI, TSN5		
 Washington @  Calgary 9:30 PM ET SNF, CSN-DC+		

Tuesday, Nov 1











Matchup	Time	Networks
 St. Louis @  NY Rangers 7:00 PM ET MSG+, FS-MW		



Create/Update Flavors

- `create(..)`
- `update(..)`
- `createOrUpdate(..)`
- `idempotentCreateOrUpdate(..)`
- `idempotentFullSync(..)`
- `delete(..)`











Demo: Live Game Events App

1st Period			
	00:23	MISSED SHOT	Wayne Simmonds - Wide of Net
	00:33	SHOT	Mark Streit Slap Shot saved by Marc-Andre Fleury
	00:44	HIT	Claude Giroux hit Scott Wilson
	00:57	BLOCKED SHOT	Brandon Manning blocked shot from Evgeni Malkin
	01:07	SHOT	Phil Kessel Wrist Shot saved by Steve Mason
	01:20	HIT	Chris Kunitz hit Radko Gudas
	01:29	SHOT	Sean Couturier Wrist Shot saved by Marc-Andre Fleury
	03:39	SHOT	Patric Hornqvist Wrist Shot saved by Steve Mason
	04:01	FACEOFF	Sean Couturier faceoff won against Evgeni Malkin
	04:03	SHOT	Travis Konecny Wrist Shot saved by Marc-Andre Fleury

Takeaway

- POJOs can be exposed via LinkRest using annotations
- Custom backend is plugged in as a "stage listener"
- "include" / "exclude" / "limit" supported automatically
- Other protocol features are up to the backend to implement

Demo: Hockey Schedule with Live Scores

 PIT 5  PHI 4 FINAL	 BOS 1  DET 0 FINAL	 TOR 1  MTL 2 FINAL	 TBL 1  NJD 3 FINAL	 DAL 0  MIN 4 FINAL
--	--	--	--	--

Takeaway

- LinkRest can do parallel fetching and aggregation from multiple data sources.

Great Minds Think Alike



Describe your data

```
type Project {  
  name: String  
  tagline: String  
  contributors: [User]  
}
```

Ask for what you want

```
{  
  project(name: "GraphQL") {  
    tagline  
  }  
}
```

Get predictable results

```
{  
  "project": {  
    "tagline": "A query language for APIs"  
  }  
}
```

Questions?

<http://linkrest.io/>

@andrus_a

