

UNIVERSITÀ DEGLI STUDI ROMA TRE

Department of Computer Science and Automation

How to Extract BGP Peering Information from the Internet Routing Registry

Giuseppe Di Battista

Tiziana Refice

Massimo Rimondini

ACM SIGCOMM 2006 – MINENET Workshop

15 September 2006

Pisa, Italy

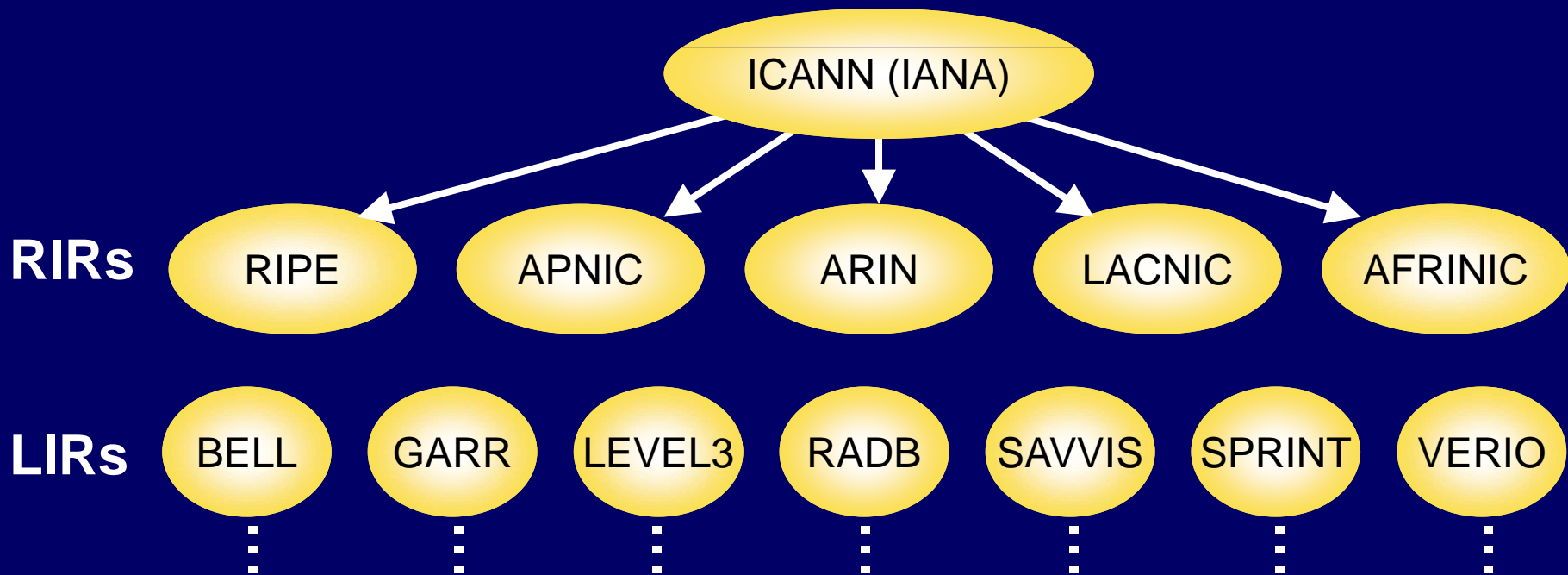


A BGP Peering



About the IRR

- Large distributed repository storing information on
 - allocation of network resources (subnets, AS #s)
 - BGP routing policies



What's the Point with the IRR?



- ✦ Born with the purpose of supporting stable and consistent routing policies
- ✦ A valuable source of information to understand Internet routing

BUT...



- ✦ Maintained on a voluntary basis
- ✦ Information is often inconsistent and/or out of date
- ✦ A systematic approach to the extraction of information is still missing

Our Contribution(s)

- ✦ A method to extract peering relationships (and more) from the IRR
- ✦ An implementation of the method as an on-line service
- ✦ A proof of effectiveness by comparison with the state of the art
- ✦ An investigation on the constructions used to specify peerings



What's Inside the IRR

- ◆ BGP routing policies described by using the **Routing Policy Specification Language (RPSL)**

values :es e

```
aut-num: AS137
import: from AS20965 action pref=100;
       from AS1299 action pref=100;
       accept ANY
export: to AS1299 announce AS-GARR
changed: noc@garr.it 20000830
source: RIPE
```

What's Inside the IRR

- ◆ BGP routing policies described by using the **Routing Policy Specification Language (RPSL)**

peering specifications nomous system

```
aut-num: AS137
import: from AS20965 action pref=100;
       from AS1299 action pref=100;
       accept ANY
export: to AS1299 announce AS-GARR
changed: noc@garr.it 20000830
source: RIPE
```

Is it That Easy?

- ◆ Structured policies
- ◆ Set objects
- ◆ Complex expressions
- ◆ Multi-protocol extensions

```
aut-num: ASX5
import: { from ASX2:AS-Z2 accept 100.0.0.0/8;
         } refine {
         from ASX1 ASX2 accept 100.1.0.0/16;
         } except {
         from ASX3 accept 100.1.1.0/24;}
export: to ASX1:PRNG-Y1
        to ASX1:AS-Z1 except ASX9
        announce 100.1.1.0/24
mp-export: to ASX11 at 2001::1 announce 2001::/48
default: to ASX12 action pref=10
default: to ASX13 100.1.1.1 at 100.1.1.2
```


Where Others Fail

- Existing tools (e.g., the **RIPE Routing Registry Consistency Check**) fail in correctly dealing with these constructions

```
aut-num: ASX5
import: { from ASX2:AS-Z2 accept 100.0.0.0/8;
        } refine {
        from ASX1 ASX2 accept 100.1.0.0/16;
        } except {
        from ASX3 accept 100.1.1.0/24;}
export: to ASX1:PRNG-Y1
        to ASX1:AS-Z1 except ASX9
        announce 100.1.1.0/24
mp-export: to ASX11 at 2001::1 announce 2001::/48
default: to ASX12 action pref=10
default: to ASX13 100.1.1.1 at 100.1.1.2
```

Is That All?

aut-num: AS24336

import: from AS17685
accept ANY

export: to AS17685
announce AS24336

changed: matsuo@po.d-b.ne.jp
20050220

source: RADB

aut-num: AS24336

import: from AS17685

import: from AS7682

export: to AS17685

export: to AS7682

changed: hm-changed@apnic.net
20050210

source: APNIC

1. Identify stubs

2. Look at the last update

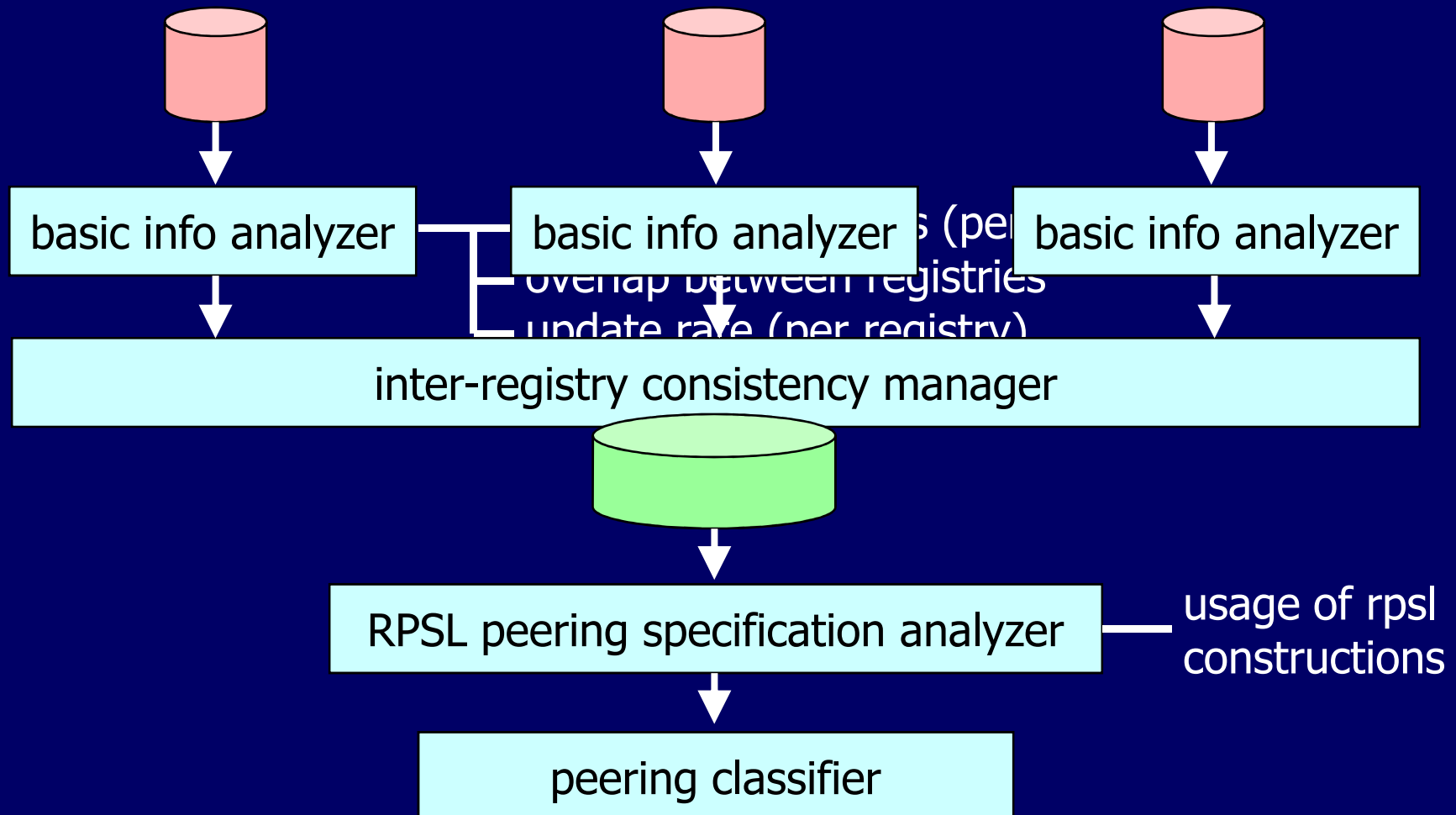
timestamp

3. Consider highest

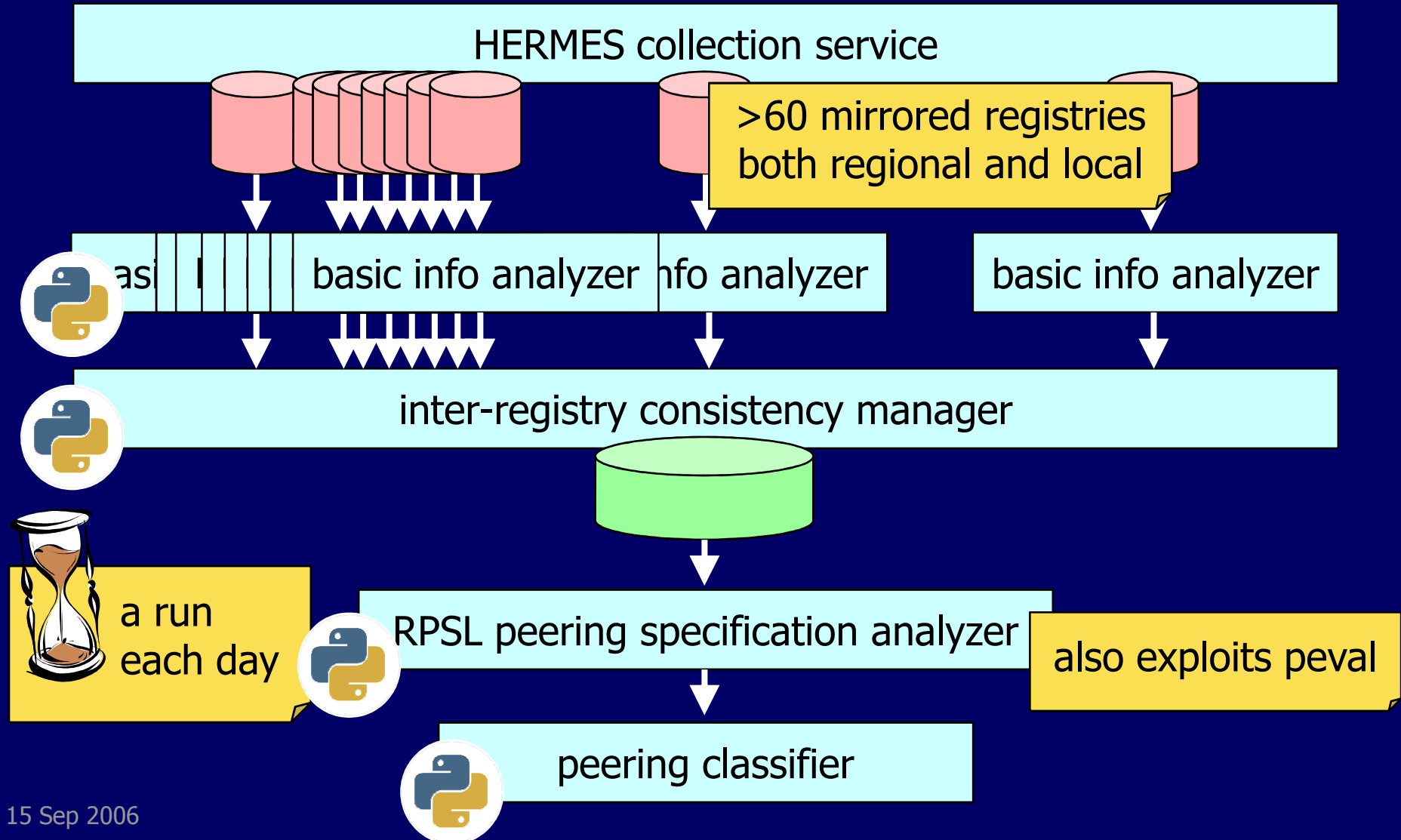
ranked (i.e. largest)

registry

An Overview of the Method



From the Method to the On-Line Service



The Peering Classifier



```
aut-num: ASX1  
import: from ASX2  
accept ASX2
```

```
aut-num: ASX2
```

- ✦ Build topologies with different levels of confidence

A Look at the Data

- ✦ 68 registries downloaded on 03/31/06
 - ftp://ftp.ripe.net/ripe/dbase
 - ftp://ftp.radb.net/radb/database
- ✦ Overlapping aut-nums

	# of aut-nums	ripe	apnic	radb	arin	verio
ripe	11468	11238	19	50	7	23
apnic	3299	19	2688	423	1	113
radb	2695	50	423	2037	37	45
arin	555	7	1	37	463	14
verio	498	23	113	45	14	310

A Look at the Data

✦ Integrating registries

definitions	aut-num	as-set	peering-set
overall	19800	7798	149
unique	18735	7478	149
potentially inconsistent	3123	306	10
potentially inconsistent	44	77	0

✦ Kept definition from highest ranked registry

A Look at the Data

	# of left after aut-running	ripe	apnic	radb	arin	verio
ripe	114682%	1208	19	50	7	23
apnic	32994%	19	2688	423	1	113
radb	26957%	50	423	2037	37	45
arin	5551%	7	1	37	463	14
verio	4982%	23	113	45	14	310

Extracted Peerings

	peerings
This work	236,663
RIPE RRCC	108,521
[1] (RIPE only)	56,949
[2] (RIPE only)	70,222
[3]	127,498

[1] P. Mahadevan et al.,
*The Internet AS-Level
Topology: Three Data
Sources and One Definitive
Metric.*

SIGCOMM Computer
Communication Review,
2006.

[2] B. Zhang et al.,
*Collecting the Internet AS-
Level Topology.*

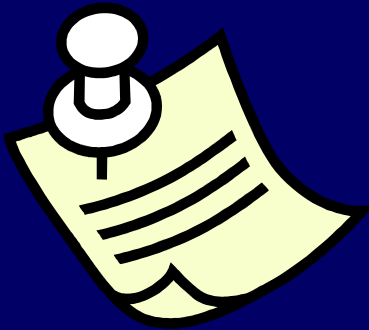
SIGCOMM Computer
Communication Review,
2005.

[3] G. Siganos et al.,
*Analyzing BGP Policies:
Methodology and Tool.*
INFOCOM 2004.

Extracted Peerings

peering type	count	from RIPE only
HALF_LEFT HALF_RIGHT	143,342	58.4%
FULL	42,599	94.6%
1/4_E	34,155	7.7%
1/4_I	13,997	23.7%
3/4_NOT_E	1,373	80.3%
3/4_NOT_I	1,013	82.2%
HALF	114	57.9%
1/2_I	51	66.7%
1/2_E	19	47.4%

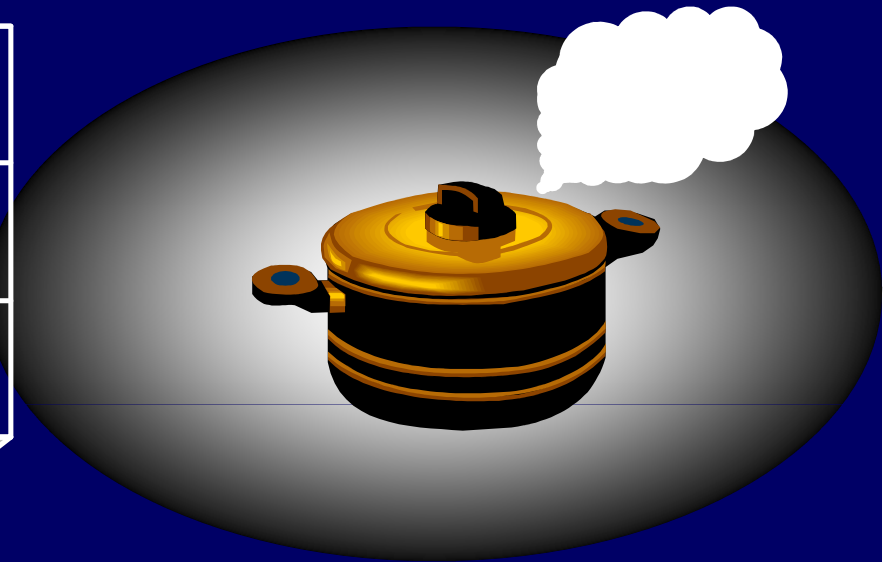
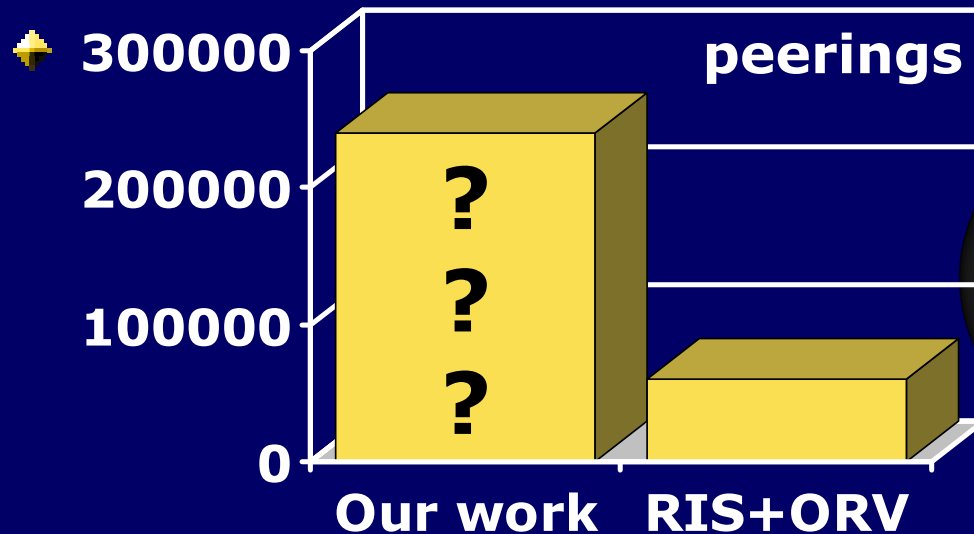
So What?



- ✦ Extracting peerings from the IRR is not trivial...
...yet it's possible
- ✦ A systematic approach
- ✦ An on-line service
- ✦ Promising results compared with past works
- ✦ Hints about the health of the IRR



In the pot...



- ◆ Who is responsible?
- ◆ Focus on BGP routing policies
- ◆ Investigate how routing information propagates
- ◆ Estimate consistency of IRR data against actual routing
- ◆ Prevent abnormal routing scenarios

Care to Have a Look?

http://tocai.dia.uniroma3.it/~irr_analysis/

Thank you!
Questions?