The technology of the IANCIS platform

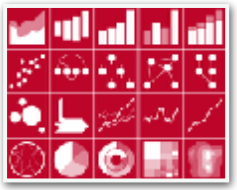
Alessandro Celestini
Institute for Applied Computing (IAC-CNR)
a.celestini@iac.cnr.it

Data Analysis Framework

ACQUISITION

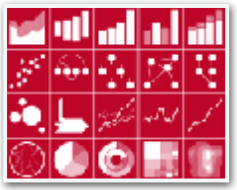
ELABORATION

VISUALIZATION



Data Analysis Framework

ACQUISITION → ELABORATION → VISUALIZATION

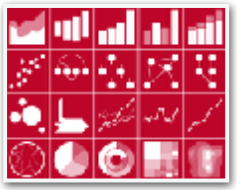


Acquisition

- *Broad* crawling
- *Focused* crawling

Data Analysis Framework

ACQUISITION → ELABORATION → VISUALIZATION



Acquisition

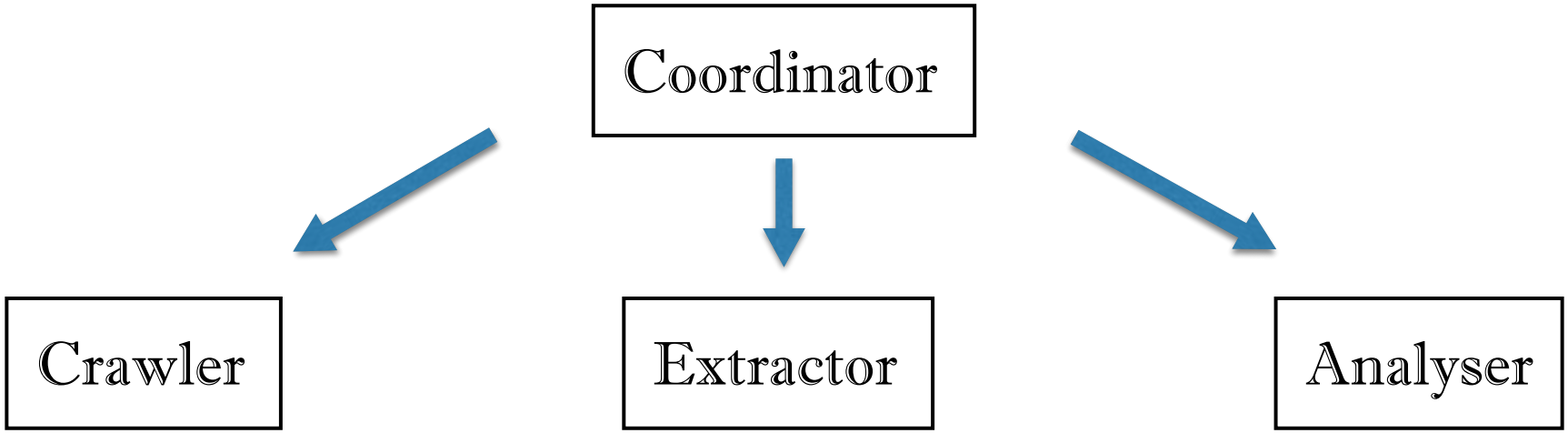
- *Broad* crawling
- *Focused* crawling

Elaboration

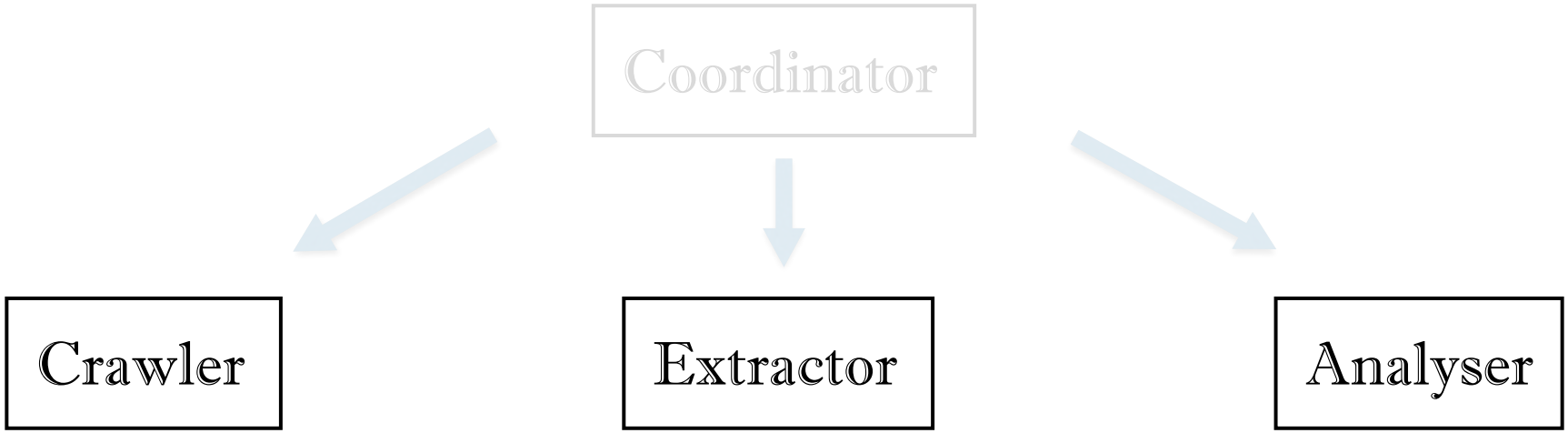
- *Data grabbing* from web pages
- *Text extraction* from web resources
- *Analysis* of extracted texts

Platform Workflow

Platform Workflow



Platform Workflow



Platform Workflow

Coordinator

Crawler

Extractor

Analyser

BUbiNG

WARC

Platform Workflow

Coordinator

Crawler

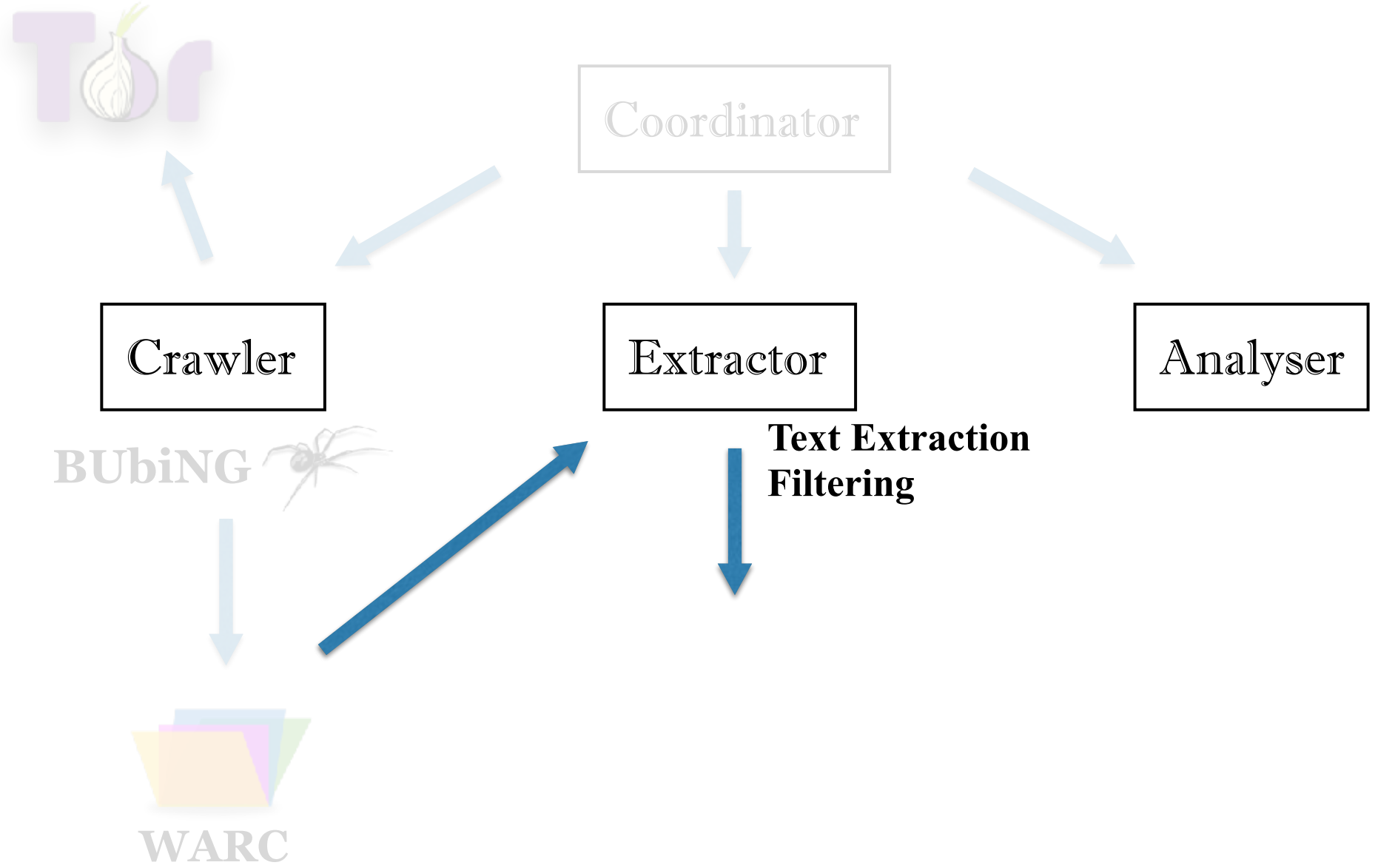
Extractor

Analyser

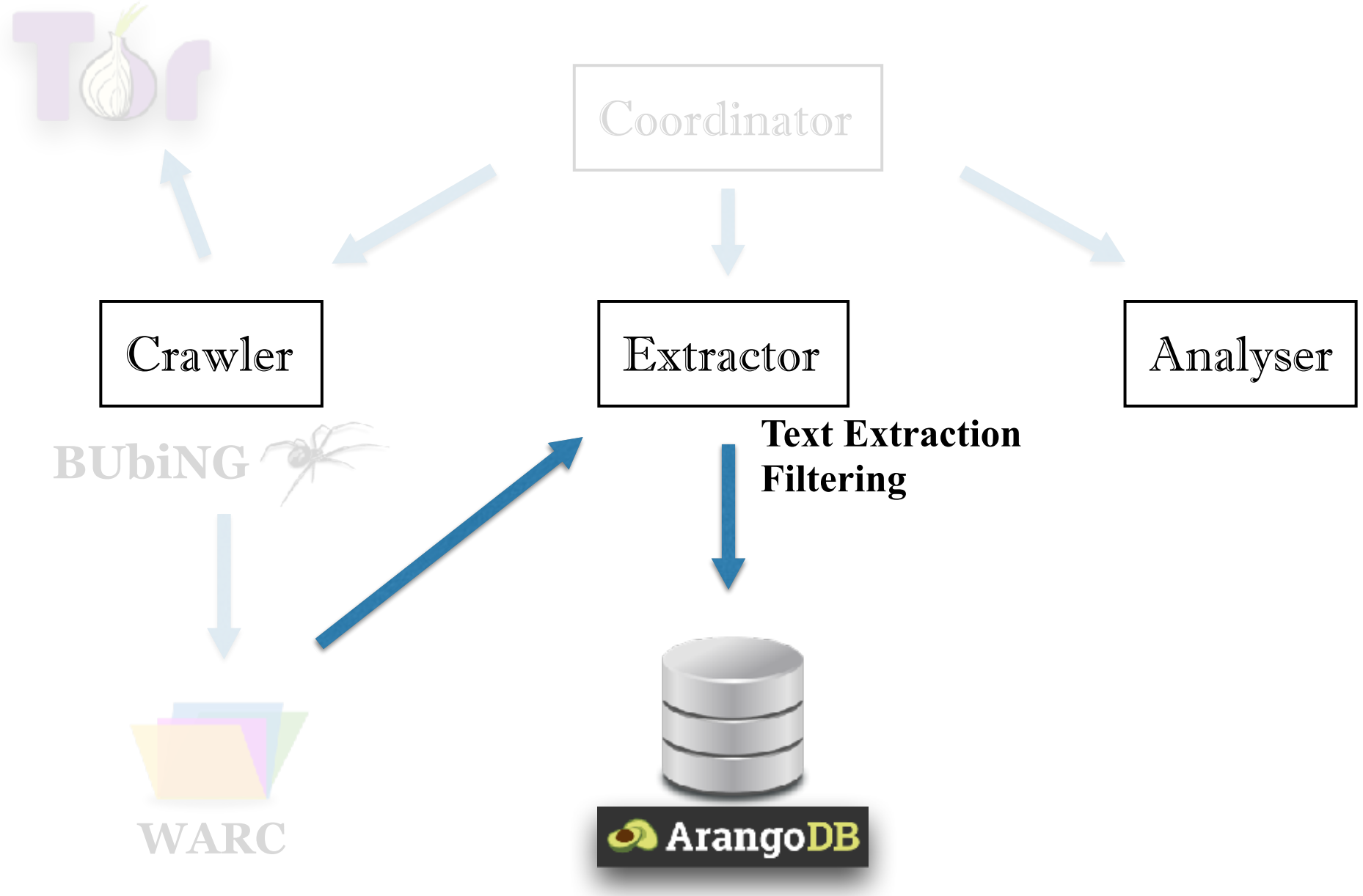
BUbiNG

WARC

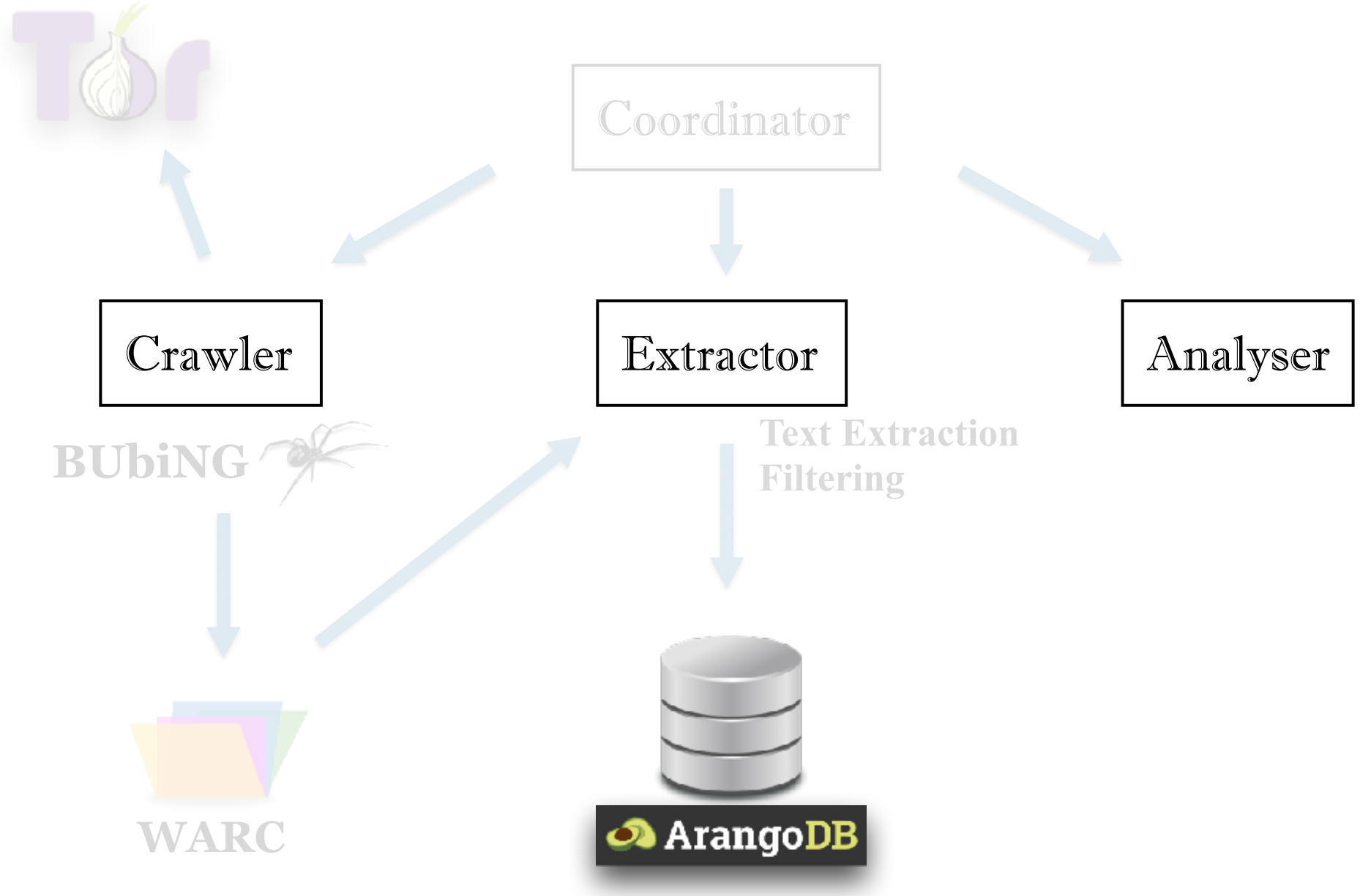
Platform Workflow



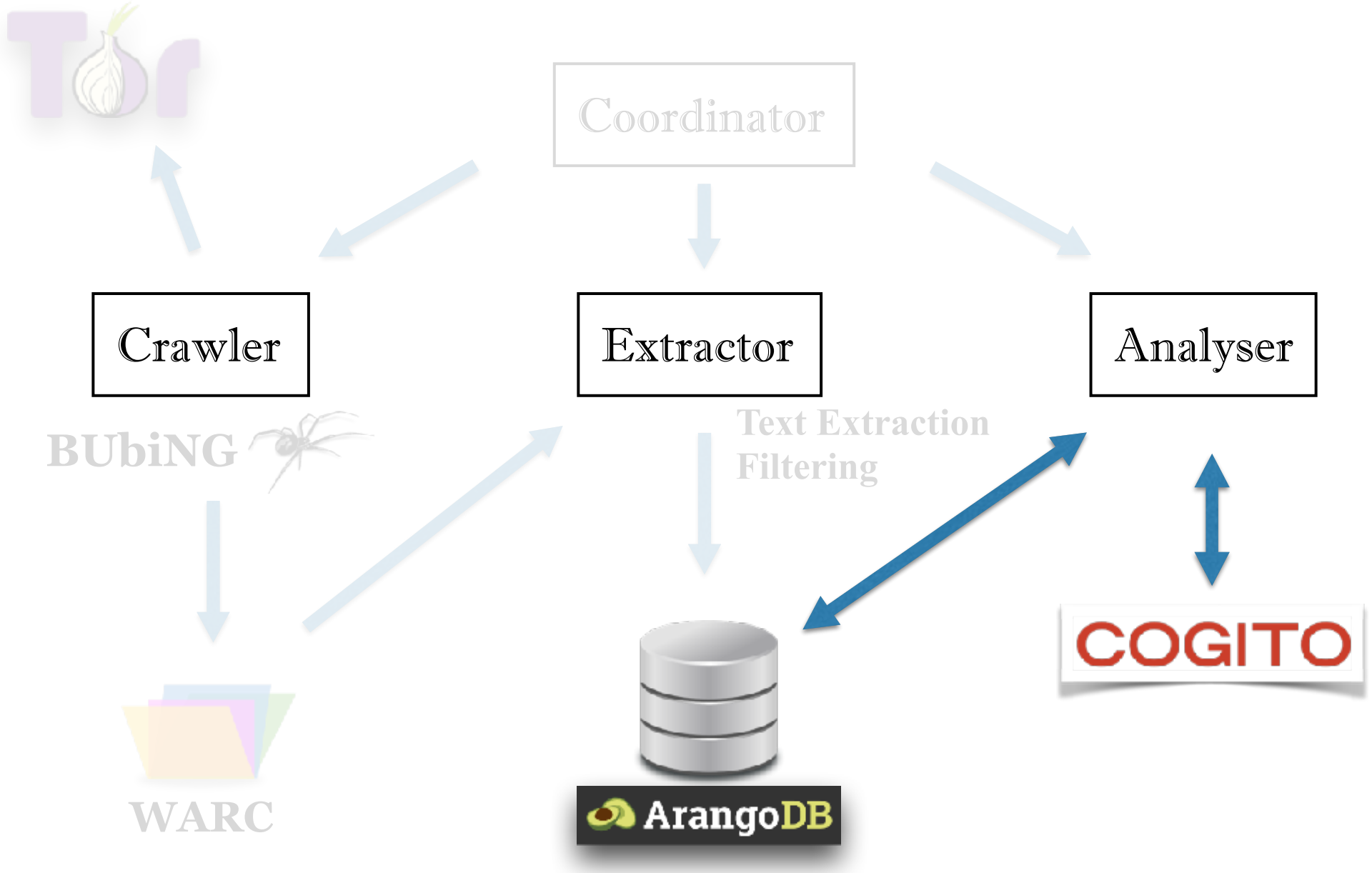
Platform Workflow



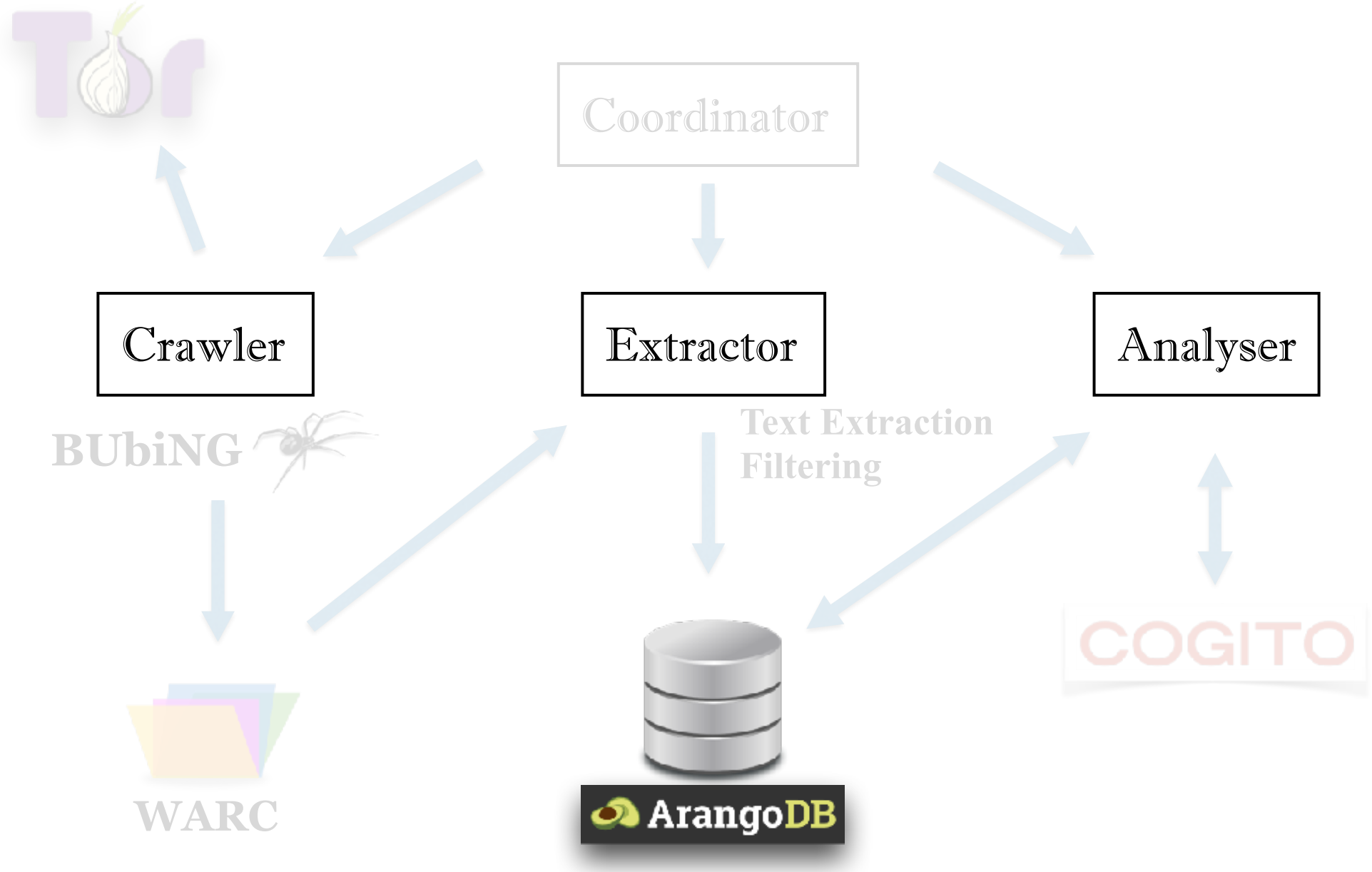
Platform Workflow



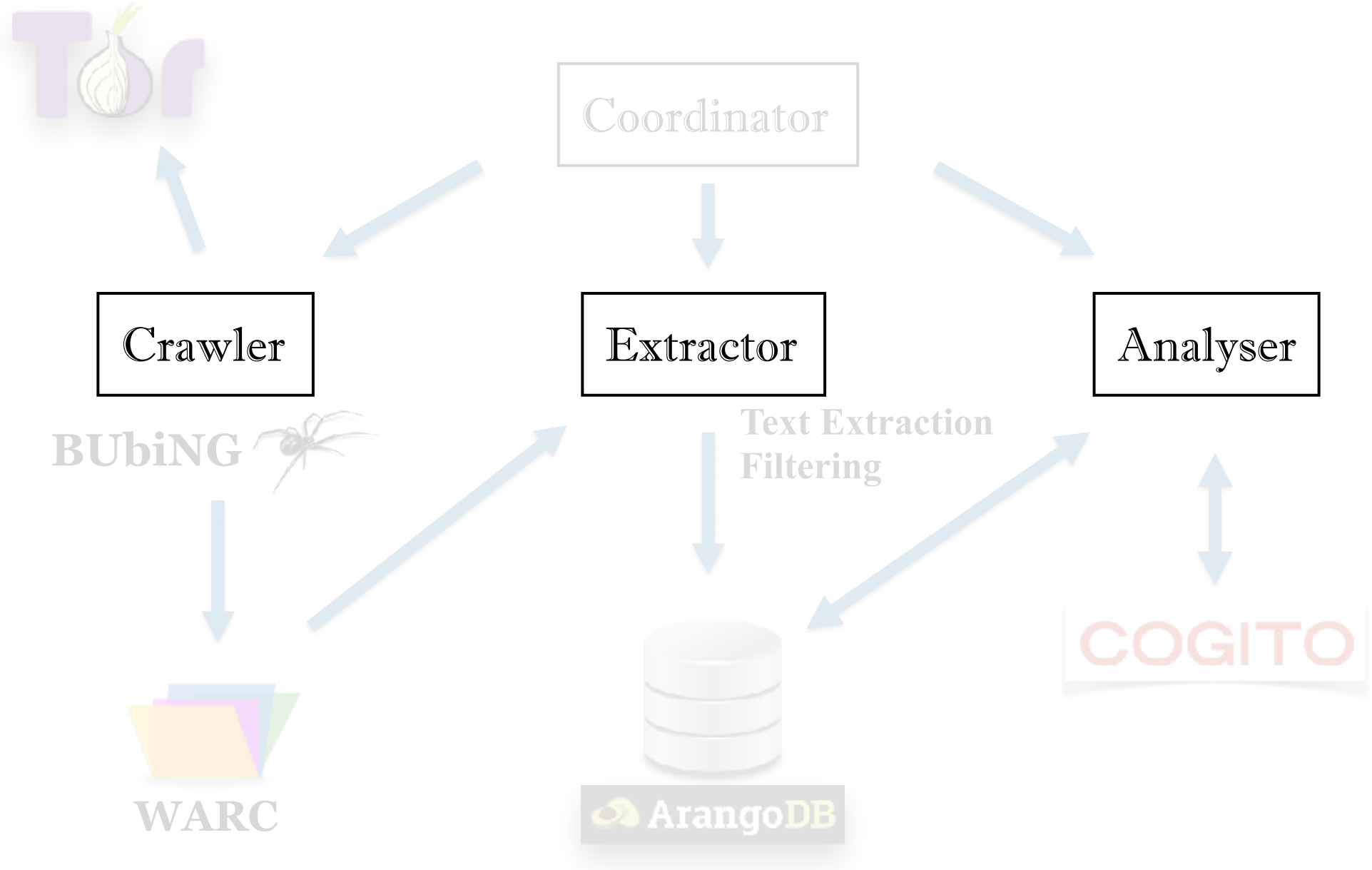
Platform Workflow



Platform Workflow



Platform Workflow



Crawling

New crawler from scratch

Explored alternatives:

BUbiNG

For our evaluation we considered: **performance, configurability, extensibility and supportability**

We tested the performance of our **crawler prototype** and evaluated the time required to implement the missing features

We identified BUbiNG as the base for our crawler component. A customized version of BUbiNG is currently part of the crawling unit of our framework

Data Storage

It supports several data models: ***documents, graphs, and key-values***

It supports ***ACID transactions*** if required

It provides a ***SQL-like query language*** or JavaScript extensions

DOCUMENTS

The storage unit is a document, in a relational database the storage unit is a record

Documents in a collection may have a ***different structure***

Data Storage

It supports several data models: ***documents, graphs, and key-values***

It supports ***ACID transactions*** if required

It provides a ***SQL-like query language*** or JavaScript extensions

DOCUMENTS

The storage unit is a document, in a relational database the storage unit is a record

Documents in a collection may have a ***different structure***

For each web resource we store the following informations

TEXT
LANGUAGE
CRAWLING_DATE
EXTRACTION_DATE

WARC_HEADER
HTTP_HEADER
TIKA_METADATA
URL

ANALYSIS_DATE
COGITO

Technologies

ACQUISITION

BUBING: high performance crawler, originally developed by the Laboratory for Web Algorithmics (LAW) of the University of Milan

Technologies

ACQUISITION

BUBING: high performance crawler, originally developed by the Laboratory for Web Algorithmics (LAW) of the University of Milan

ELABORATION **Text Extraction**

TIKA: open-source software suite for the identification and extraction of text from more than 1500 different file types

ELABORATION **Text Analysis**

COGITO: semantical analysis engine by Expert System, which classifies a text according to a suitable taxonomy, providing both quantitative and qualitative information (*what topic* is the text about and *to what extent* the text discusses such topic)

Technologies

ACQUISITION

BUBING: high performance crawler, originally developed by the Laboratory for Web Algorithmics (LAW) of the University of Milan

ELABORATION **Text Extraction**

Apache Tika

TIKA: open-source software suite for the identification and extraction of text from more than 1500 different file types

ELABORATION **Text Analysis**

COGITO: semantical analysis engine by Expert System, which classifies a text according to a suitable taxonomy, providing both quantitative and qualitative information (*what topic* is the text about and *to what extent* the text discusses such topic)

DATA STORAGE

ArangoDB

ArangoDB: is a multi-model, open-source, NoSQL database with flexible data models for documents, graphs, and key-values

Focused Crawling

OBJECTIVE:

Crawling of a known subset of websites (e.g. blogs, forums, marketplaces, etc.), and possibly only of a specific portion of such websites

CUSTOM SPIDERS

SCRAPY: an open source and collaborative application framework for crawling web sites, which can be used for a wide range of useful applications

Each web site has its own page layout, which can be used:

- to drive spiders
- to automatically grab data

Focused Crawling

SPIDERS

Each spider crawls a targeted website

TOR WEBSITES

Focused Crawling

SPIDERS

Each spider crawls a targeted website

TOR WEBSITES

Relevant data are grabbed from each page

DATA GRABBING

Data are stored and organized in a database

DATA STORAGE

Focused Crawling

SPIDERS

Each spider crawls a targeted website

TOR WEBSITES

Relevant data are grabbed from each page

DATA GRABBING

Data are stored and organized in a database

DATA STORAGE

SEMI-STRUCTURED data *(web pages)* are transformed into **STRUCTURED** data *(DB tables)*

Exploring and Analyzing the Tor Hidden Services Graph

Analyzing the Tor Graph

OBJECTIVE:

Gather ***information concerning the topology*** of the Tor Web and identify potential relationships among ***topological properties and hidden services' contents***

Analyzing the Tor Graph

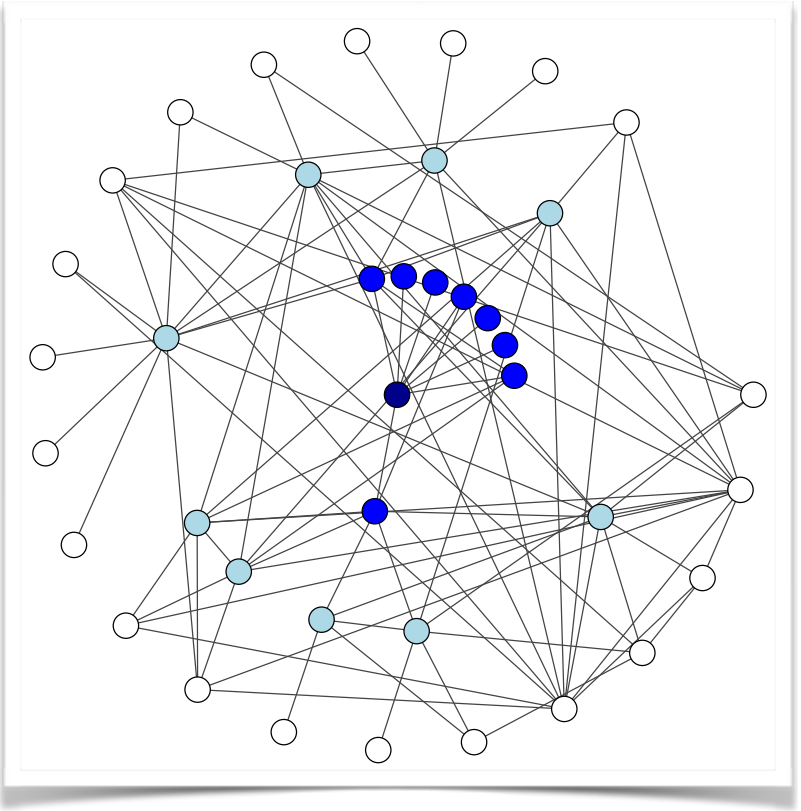
OBJECTIVE:

Gather ***information concerning the topology*** of the Tor Web and identify potential relationships among ***topological properties and hidden services' contents***

We construct 3 graphs:

- Page Graph (PG)
- Host Graph (HG)
- Service Graph (SG)

Graph	# Nodes	# Arcs
Page Graph	918885	17963147
Host Graph	5420	64379
Service Graph	5144	59492



Analyzing the Tor Graph

TOPOLOGY

We identify ***the most important nodes*** in the network, namely those with highest degrees and Betweenness Centrality*

Top nodes represents interesting ***targets for investigation***

Table IV: Degree and Betweenness Centrality (Top 10)

Rank	Node Index				
	In-degree (PG)	Out-degree (PG)	Degree (PG*)	BC (PG)	BC (PG*)
1	255	6855	255	142	154
2	3399	496	496	452	461
3	13499	8107	3399	653	4217
4	341315	42224	13499	422	15324
5	344053	218797	341315	12630	28608
6	341666	103530	343628	3467	37137
7	790729	131172	344053	1558	39458
8	343628	130099	341666	348	40354
9	342870	128671	790729	1913	42400
10	411	128096	6855	1198	46429

*BC measures the extent to which a vertex lies on paths between other vertices

Analyzing the Tor Graph

TOPOLOGY

We identify ***the most important nodes*** in the network, namely those with highest degrees and Betweenness Centrality*

Top nodes represents interesting ***targets for investigation***

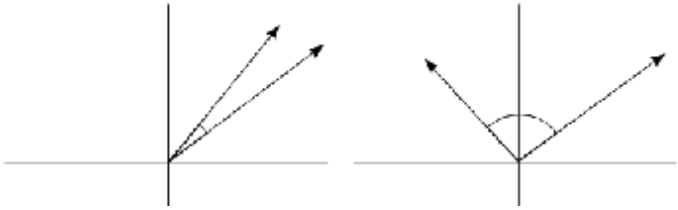
Table IV: Degree and Betweenness Centrality (Top 10)

Rank	Node Index				
	In-degree (PG)	Out-degree (PG)	Degree (PG*)	BC (PG)	BC (PG*)
1	255	6855	255	142	154
2	3399	496	496	452	461
3	13499	8107	3399	653	4217
4	341315	42224	13499	422	15324
5	344053	218797	341315	12630	28608
6	341666	103530	343628	3467	37137
7	790729	131172	344053	1558	39458
8	343628	130099	341666	348	40354
9	342870	128671	790729	1913	42400
10	411	128096	6855	1198	46429

HIDDEN SERVICES' CONTENTS

We use ***Cogito's output to represent the content*** of each visited page (a vector for each page)

We use the ***cosine similarity*** to measure the similarity between any two pages



*BC measures the extent to which a vertex lies on paths between other vertices

Analyzing the Tor Graph

FINDING COMMUNITIES:

We can identify ***components with increasing semantic similarity by cutting off graph nodes with increasing BC score*** (or increasing degree)

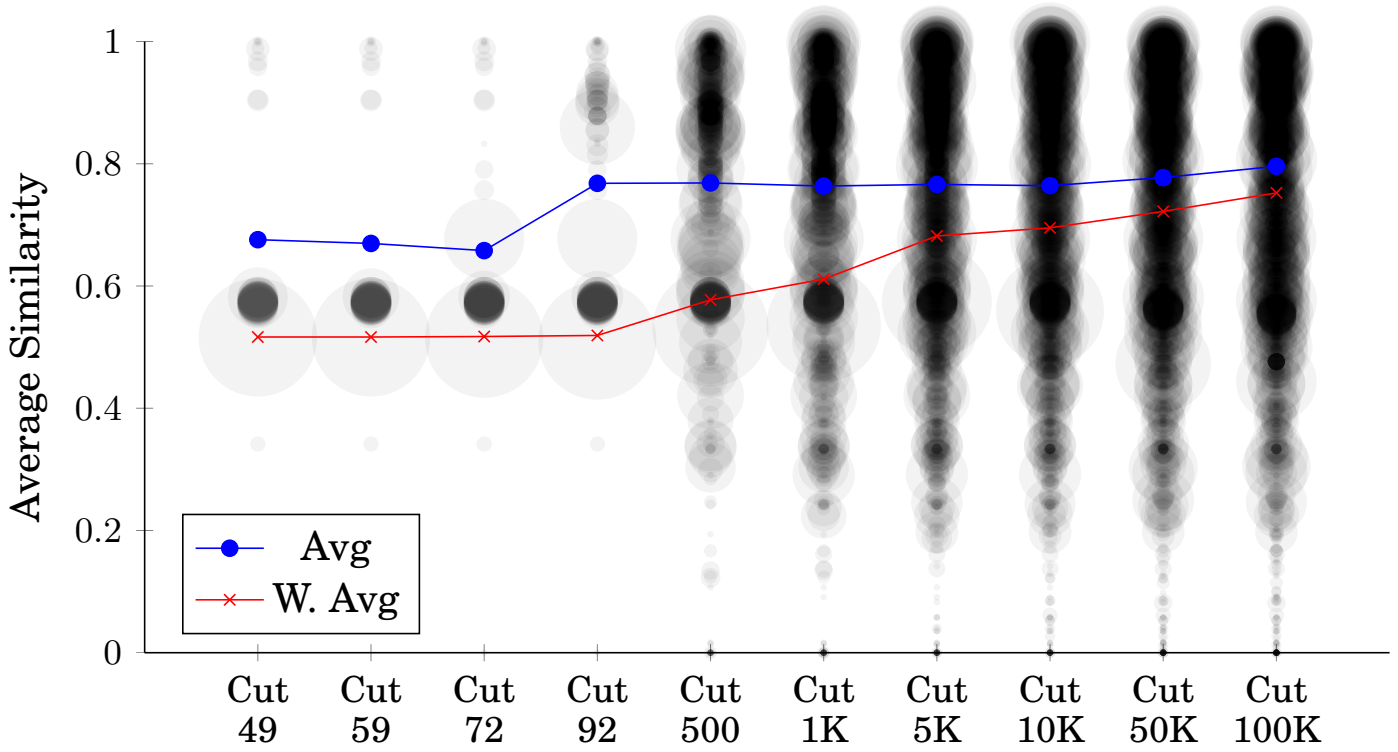
It emerges that the ***BC is a better performing metrics than the degree*** to the end of identifying semantic uniform sub-components

Analyzing the Tor Graph

FINDING COMMUNITIES:

We can identify ***components with increasing semantic similarity by cutting off graph nodes with increasing BC score*** (or increasing degree)

It emerges that the ***BC is a better performing metrics than the degree*** to the end of identifying semantic uniform sub-components



Thank you for your time