Übersicht: Open Source Webspider Heritrix

Dr. Christian Herta

June 14, 2009

Einführung

- *Heritrix*[1] ist ein veralteres englisches Wort für Erbin
- Lizenz: LGPL (gnu lesser general public licence)
- Ziel: allgemeines Framework zum Spidern mit austauschbare Komponenten
  - Standard-Komponenten enthalten

Recap: Logische Schritte beim Spidern

Recap: Logische Schritte beim Spidern

- Wähle eine URI aus der URI-Liste (*frontier*) aus

Recap: Logische Schritte beim Spidern

- Wähle eine URI aus der URI-Liste (*frontier*) aus
- Hole (*Fetch*) die URI

Recap: Logische Schritte beim Spidern

- Wähle eine URI aus der URI-Liste (*frontier*) aus
- Hole (*Fetch*) die URI
- Index

Recap: Logische Schritte beim Spidern

- Wähle eine URI aus der URI-Liste (*frontier*) aus
- Hole (*Fetch*) die URI
- Index
- Füge die ausgewählten, extrahierten Link-URIs der URI-Liste hinzu

Recap: Logische Schritte beim Spidern

- Wähle eine URI aus der URI-Liste (*frontier*) aus
- Hole (*Fetch*) die URI
- Index
- Füge die ausgewählten, extrahierten Link-URIs der URI-Liste hinzu
- Notiere, dass die URI verarbeitet wurde

Wichtigste Bestandteile

- *Scope*: Seeds und Auswahl/Filter-Regeln zu den URIs

Wichtigste Bestandteile

- *Scope*: Seeds und Auswahl/Filter-Regeln zu den URIs
- *Processor Chains*: URI- und Dokumentenverarbeitung, unter Anderem für:

Wichtigste Bestandteile

- *Scope*: Seeds und Auswahl/Filter-Regeln zu den URIs
- *Processor Chains*: URI- und Dokumentenverarbeitung, unter Anderem für:
  - Holen der IPs (*DNS*) und Seiten

Wichtigste Bestandteile

- *Scope*: Seeds und Auswahl/Filter-Regeln zu den URIs
- *Processor Chains*: URI- und Dokumentenverarbeitung, unter Anderem für:
  - Holen der IPs (*DNS*) und Seiten
  - Extraktion der Links

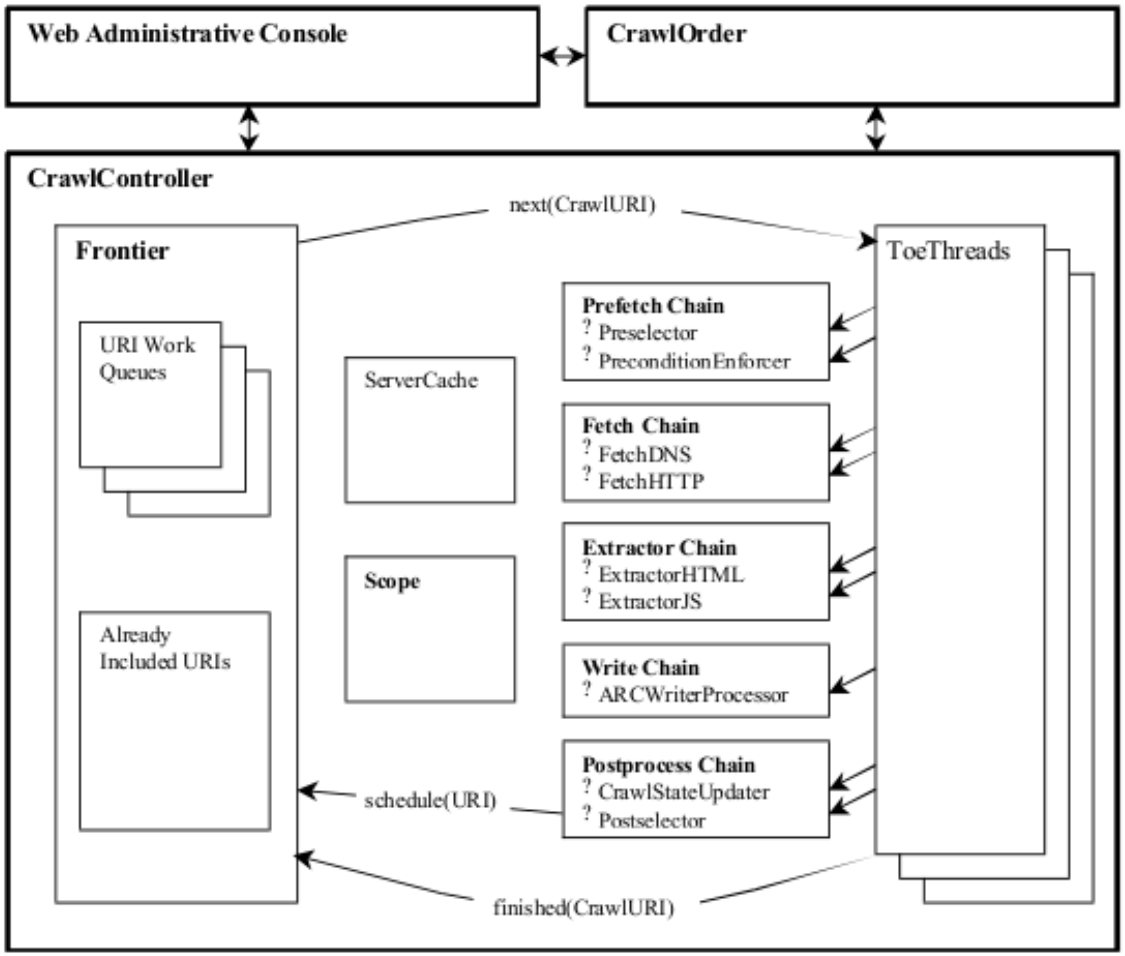
Wichtigste Bestandteile

- *Scope*: Seeds und Auswahl/Filter-Regeln zu den URIs
- *Processor Chains*: URI- und Dokumentenverarbeitung, unter Anderem für:
  - Holen der IPs (*DNS*) und Seiten
  - Extraktion der Links
  - Filtern und Normalisierung der URIs

Wichtigste Bestandteile

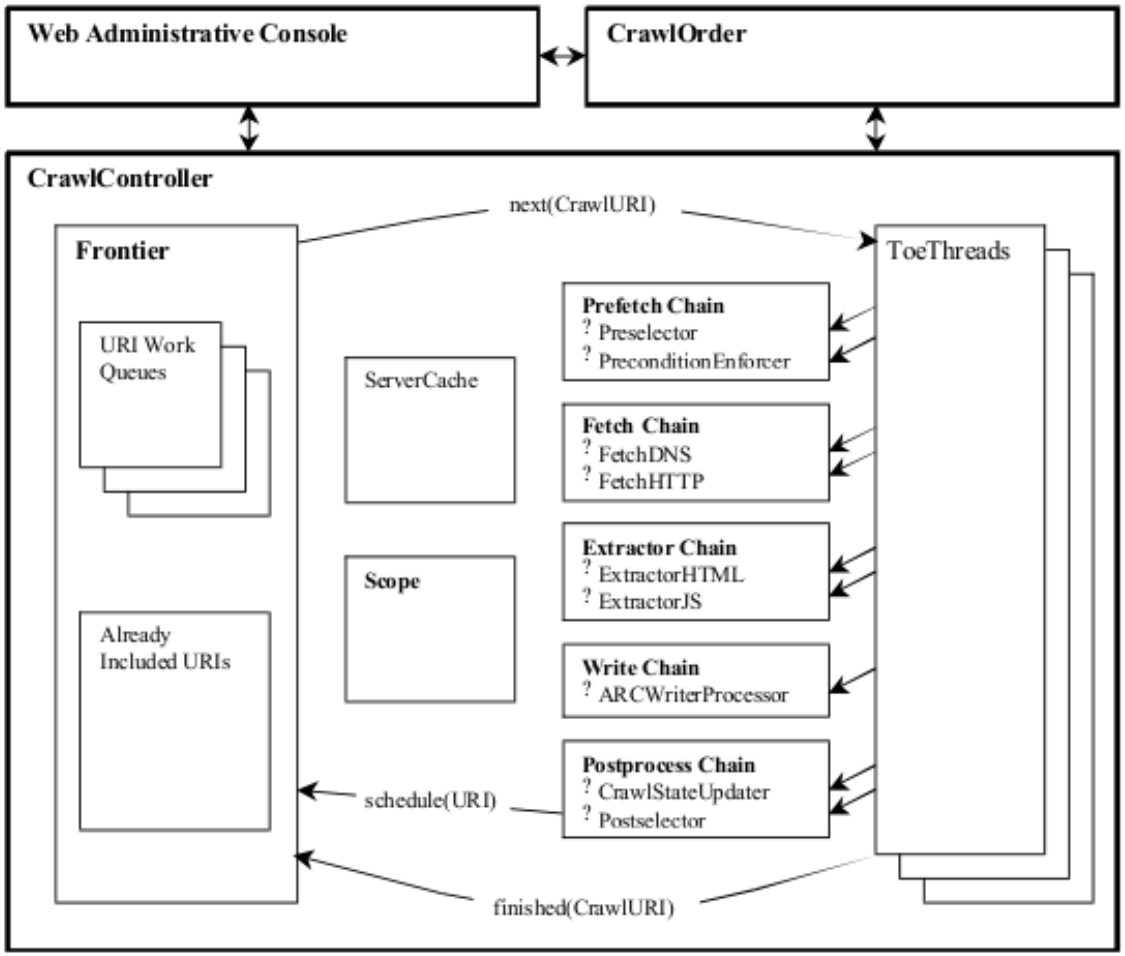
- *Scope*: Seeds und Auswahl/Filter-Regeln zu den URIs
- *Processor Chains*: URI- und Dokumentenverarbeitung, unter Anderem für:
  - Holen der IPs (*DNS*) und Seiten
  - Extraktion der Links
  - Filtern und Normalisierung der URIs
- *Frontier*

Übersicht aus [1]

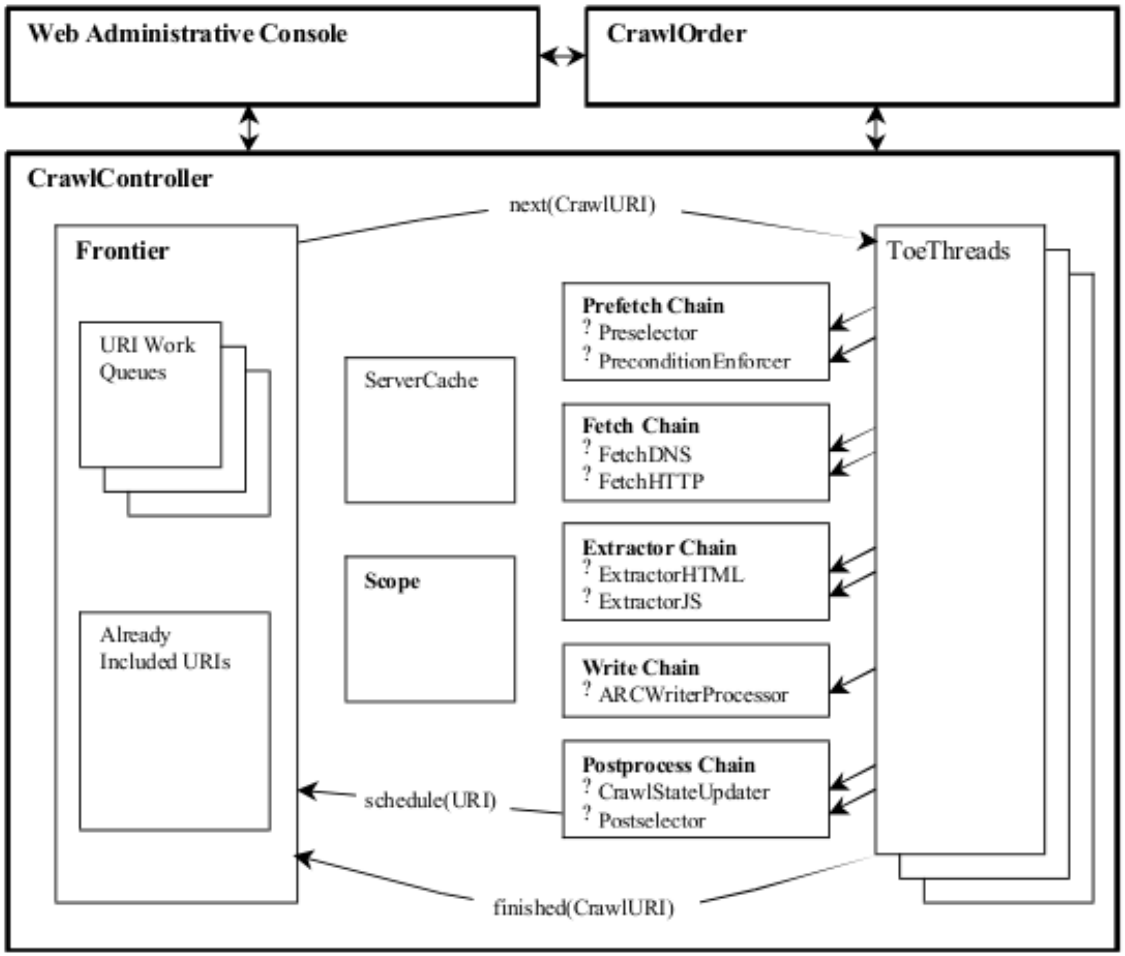


Übersicht aus [1]

- Web Administrative Console (Web-GUI zur Konfiguration)

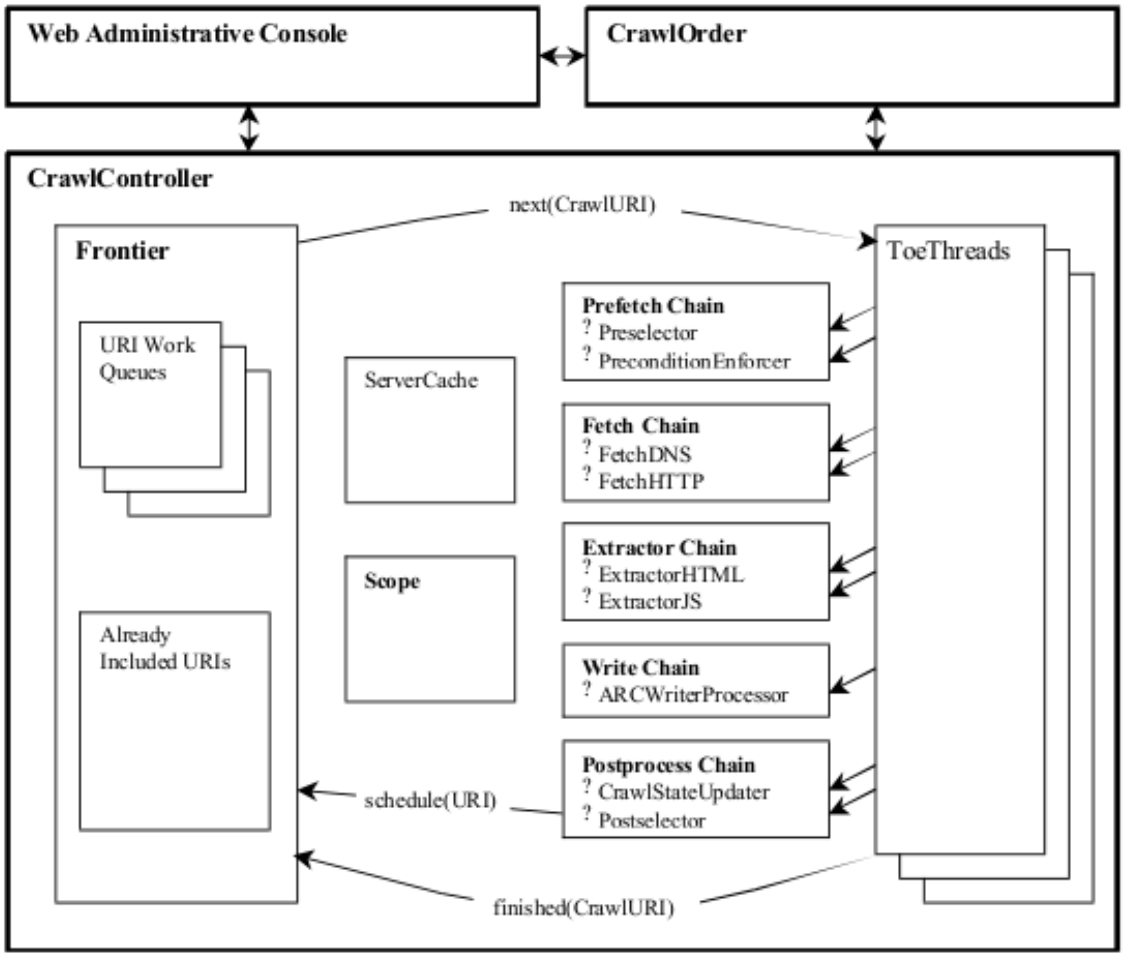


Übersicht aus [1]



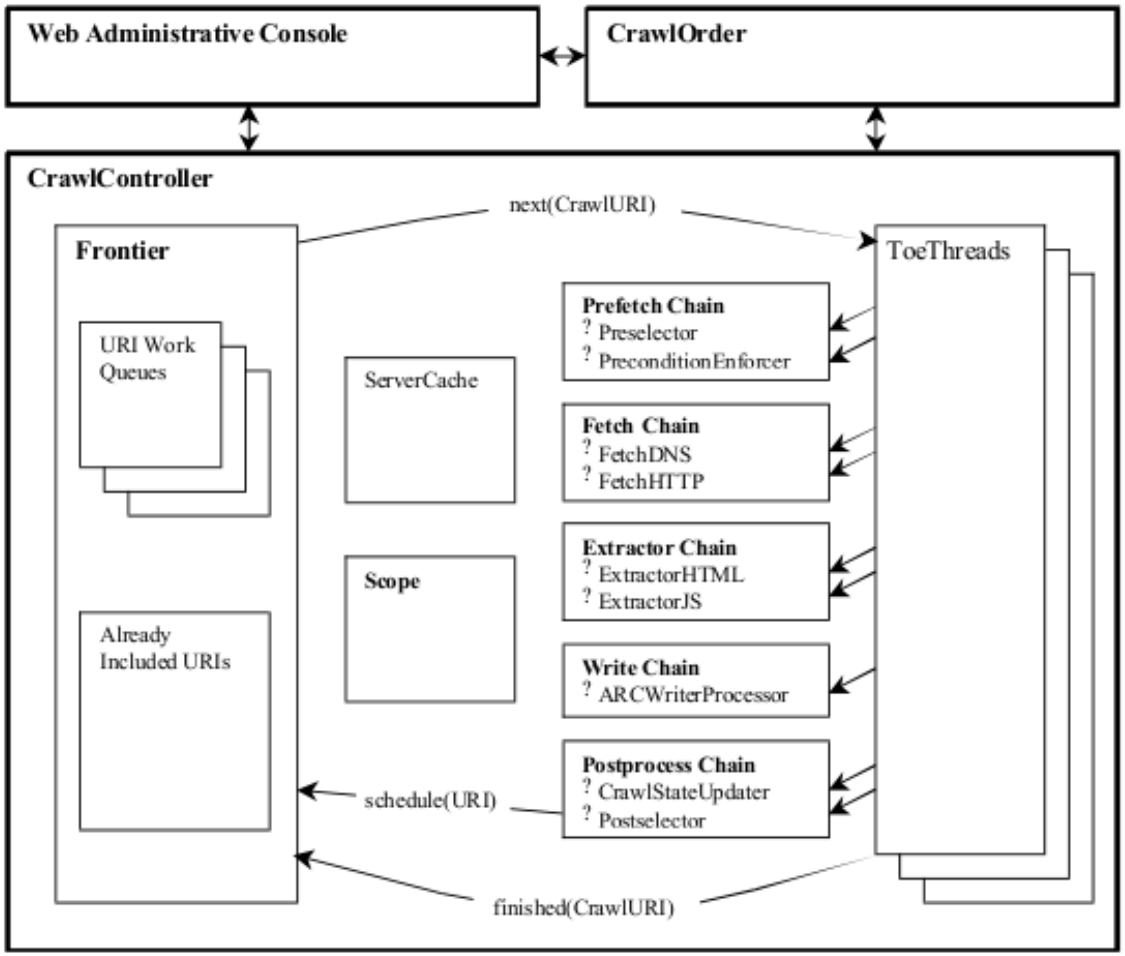
- Web Administrative Console (Web-GUI zur Konfiguration)
- CrawlOrder (Konfigurations-Objekt - externe XML-Repräsentation)

Übersicht aus [1]



- Web Administrative Console (Web-GUI zur Konfiguration)
- CrawlOrder (Konfigurations-Objekt - externe XML-Repräsentation)
- Crawl-Controller: mit Referenzen zu allen Crawl-Komponenten

Übersicht aus [1]



- Web Administrative Console (Web-GUI zur Konfiguration)
- CrawlOrder (Konfigurations-Objekt - externe XML-Repräsentation)
- Crawl-Controller: mit Referenzen zu allen Crawl-Komponenten
- Scope: Initiale "Füttern" der Frontier und Filterregeln

Threads

- Multithreaded

Threads

- Multithreaded
- *Worker threads* heißen: *ToeThreads*
  - Frage die *Frontier* nach der nächsten URI

Threads

- Multithreaded
- *Worker threads* heißen: *ToeThreads*
  - Frage die *Frontier* nach der nächsten URI
  - Reiche die URI durch die Prozessoren durch

Threads

- Multithreaded
- *Worker threads* heißen: *ToeThreads*
  - Frage die *Frontier* nach der nächsten URI
  - Reiche die URI durch die Prozessoren durch
  - Reporte *finished()* der URI

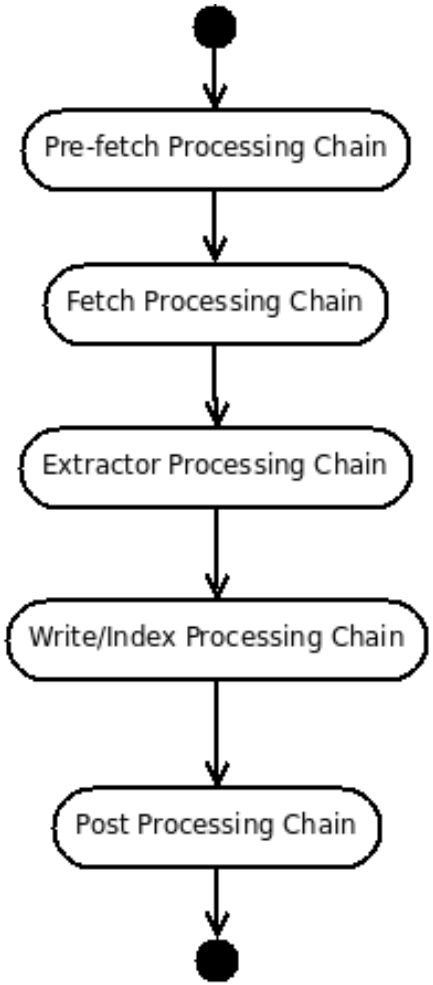
Threads

- Multithreaded
- *Worker threads* heißen: *ToeThreads*
  - Frage die *Frontier* nach der nächsten URI
  - Reiche die URI durch die Prozessoren durch
  - Reporte *finished()* der URI
- Größenordung der ToeThreads $\approx 10^2$

URIs und Server Repräsentation

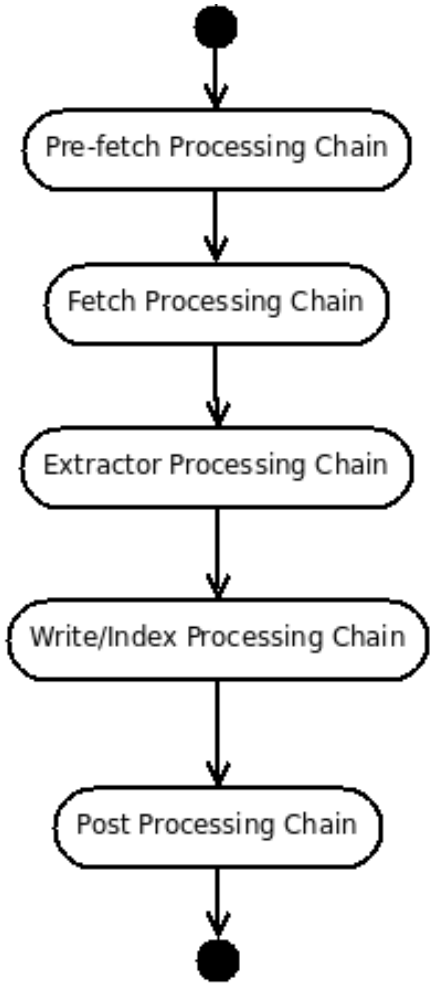
- *Server-Cache* hält Instanzen von *CrawlServer*-Instanzen. Diese speichern Information über
  - IP Adressen
  - *robots exclusion policies*,
  - *reponsiveness*
  - *per-host crawl* Statistiken
- *CrawlURI*-Instanz repräsentiert URI
- Verhalten des Crawlers wird stark bestimmt durch die verwendeten und konfigurierten Prozesssoren

Fünf Arten von Prozessor-Typen und *Processor Chains*



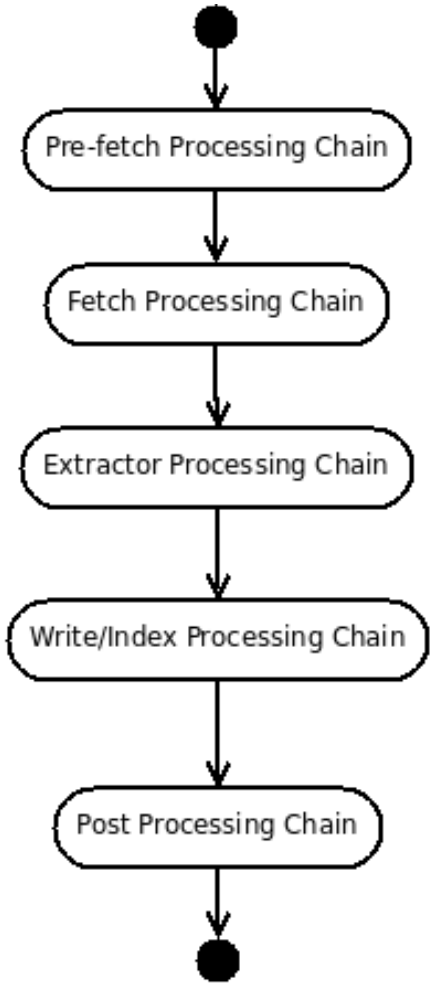
Fünf Arten von Prozessor-Typen und *Processor Chains*

1. *Prefetch Chain*: vor jeglicher Netzwerk-Aktivität,z.B. Gewährleistung der Berücksichtigung der `robots.txt`(*Fetch*, *Considering*); *delay, reorder or veto the subsequent processing of a CrawlURIs*



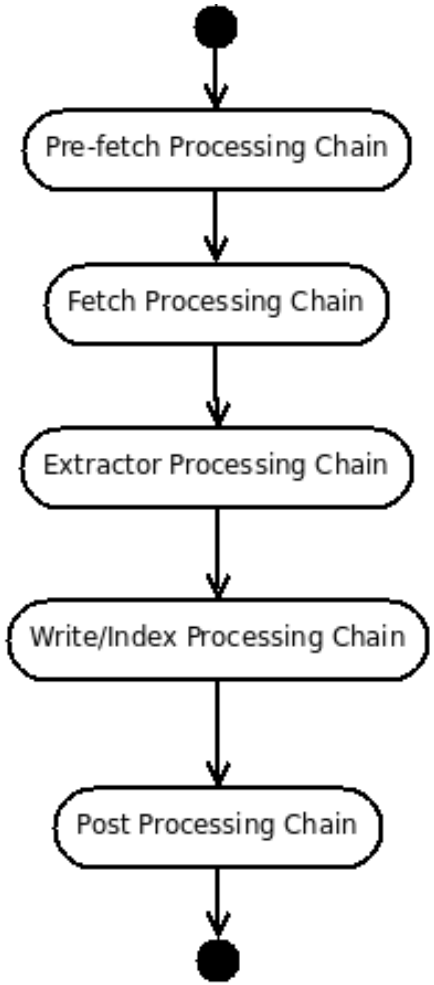
Fünf Arten von Prozessor-Typen und *Processor Chains*

1. *Prefetch Chain*: vor jeglicher Netzwerk-Aktivität,z.B. Gewährleistung der Berücksichtigung der `robots.txt`(*Fetch*, *Considering*); *delay, reorder or veto the subsequent processing of a CrawlURIs*
2. *Fetch Chain*: Netzwerk-Aktivität



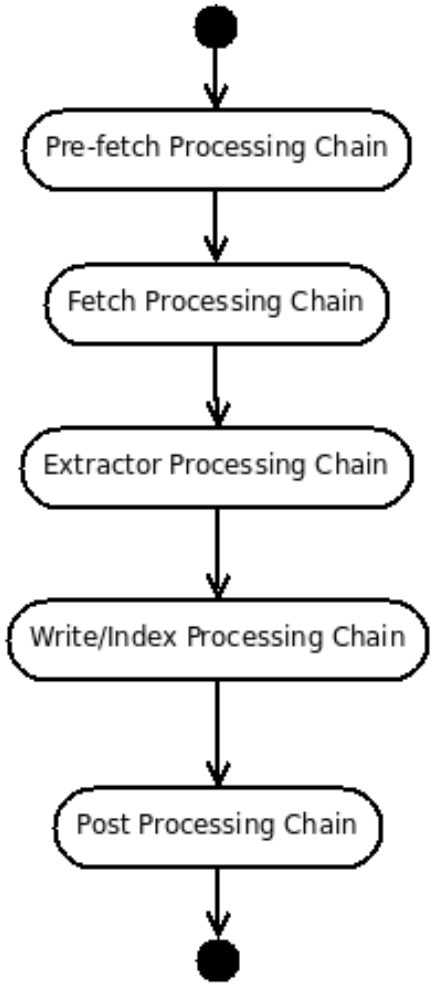
Fünf Arten von Prozessor-Typen und *Processor Chains*

1. *Prefetch Chain*: vor jeglicher Netzwerk-Aktivität,z.B. Gewährleistung der Berücksichtigung der `robots.txt`(*Fetch*, *Considering*); *delay, reorder or veto the subsequent processing of a CrawlURIs*
2. *Fetch Chain*: Netzwerk-Aktivität
3. *Extract Chain*: Extraktion von *features of interest*



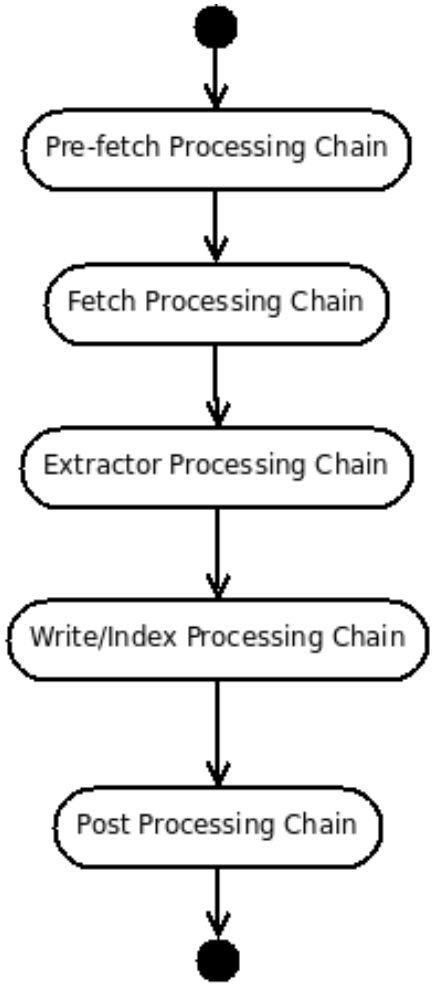
Fünf Arten von Prozessor-Typen und *Processor Chains*

1. *Prefetch Chain*: vor jeglicher Netzwerk-Aktivität,z.B. Gewährleistung der Berücksichtigung der `robots.txt`(*Fetch*, *Considering*); *delay, reorder or veto the subsequent processing of a CrawlURIs*
2. *Fetch Chain*: Netzwerk-Aktivität
3. *Extract Chain*: Extraktion von *features of interest*
4. *Write Chain*: Speichern des Crawl-Ergebniss (z.B. *Content* und URIs)



Fünf Arten von Prozessor-Typen und *Processor Chains*

1. *Prefetch Chain*: vor jeglicher Netzwerk-Aktivität,z.B. Gewährleistung der Berücksichtigung der `robots.txt`(*Fetch*, *Considering*); *delay, reorder or veto the subsequent processing of a CrawlURIs*
2. *Fetch Chain*: Netzwerk-Aktivität
3. *Extract Chain*: Extraktion von *features of interest*
4. *Write Chain*: Speichern des Crawl-Ergebniss (z.B. *Content* und URIs)
5. *Postprocess Chain*: URI-Filtern zur Berücksichtigung des Scopes, Füttern der Frontier



Wichtige Prozessor-Module aus [1]

	Name	Function
Prefetch	Preselector	Offers an opportunity to reject previously-scheduled URIs not of interest.
	PreconditionEnforcer	Ensures that any URIs which are preconditions for the current URI are scheduled beforehand.
Fetch	FetchDNS	Performs DNS lookups, for URIs of the "*dns:*" scheme.
	FetchHTTP	Performs HTTP retrievals, for URIs of the "*http:*" and "*https:*" schemes.
Extract	ExtractorHTML	Discovers URIs inside HTML resources.
	ExtractorJS	Discovers likely URIs inside Javascript resources.
	ExtractorCSS	Discovers URIs inside Cascading Style Sheet resources.
	ExtractorSWF	Discovers URIs inside Shockwave/Flash resources.
	ExtractorPDF	Discovers URIs inside Adobe Portable Document Format resources.
	ExtractorDOC	Discovers URIs inside Microsoft Word document resources.
	ExtractorUniversal	Discovers legal URI patterns inside any resource with an ASCII-like encoding.
Write	ARCWriterProcessor	Writes retrieved resources to a series of files in the Internet Archive's ARC file format.
Postprocess	Postselector	Evaluates URIs discovered by previous processors against the configured crawl Scope, scheduling those of interest to the Frontier.
	CrawlStateUpdater	Updates crawler-internal caches with new information retrieved by earlier processors.

Benutzung von Heritrix

mittels Web-GUI in der Vorlesung


G. Mohr, M. Stack, I. Ranitovic, D. Avery, and M. Kimpton.
An introduction to heritrix.
Proceedings of IWAW'04, 2004.