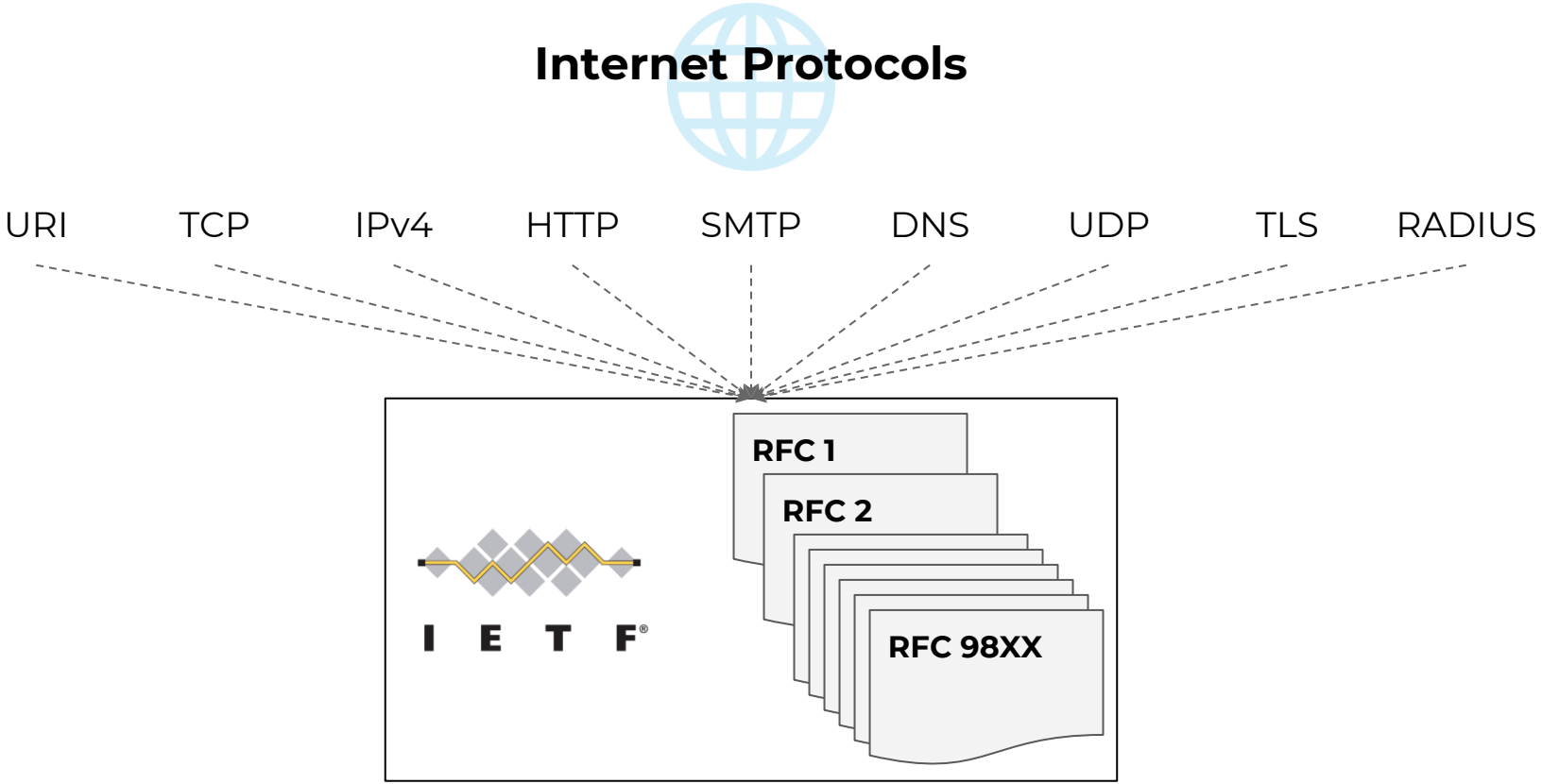
RFCScope: Detecting Logical Ambiguities in Internet Protocol Specifications

Mrigank Pawagi[1] Lize Shao[2] Hyeonmin Lee[2]
Yixin Sun[2] Wenxi Wang[2]

[1]*Indian Institute of Science* [2]*University of Virginia*

Introduction



Introduction

- Natural Language (in RFC 9460)

Within a SVCB RRset, all RRs SHOULD have the same mode. If an RRset contains a record in AliasMode, the recipient MUST ignore any ServiceMode records in the set.

…

- Pseudocode (in RFC 9000)

```
ReadVarint(data):
 // The length of variable-length
    integers is encoded in the
 // first two bits of the first byte.
  v = data.next_byte()
  prefix = v >> 6
  length = 1 << prefix
```

…

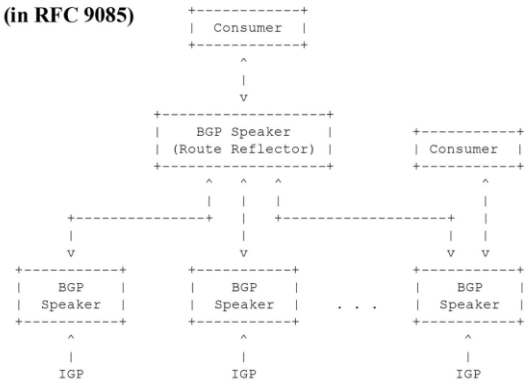
- Formal Notation (in RFC 8461)

The formal definition of the "_mta-sts" TXT record, defined using ABNF[RFC7405], is as follows:

```
 sts-text-record  =  sts-version 1*(sts-field-delim sts-field)
                     [sts-field-delim]
 sts-field        =  sts-id /           ; Note that sts-id record
                     sts-extension      ; is required.
```

…

- Diagram (in RFC 9085)



Human-written ⚠

Prone to ambiguities!

Introduction

- Natural Language (in RFC 9460)

Within a SVCB RRset, all RRs SHOULD have the same mode. If an RRset contains a record in AliasMode, the recipient MUST ignore any ServiceMode records in the set.

…

- Pseudocode (in RFC 9000)

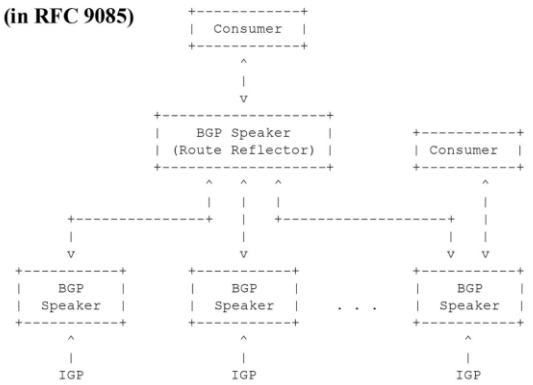
```
ReadVarint(data):
 // The length of variable-length
    integers is encoded in the
 // first two bits of the first byte.
  v = data.next_byte()
  prefix = v >> 6
  length = 1 << prefix
                …
```

- Formal Notation (in RFC 8461)

The formal definition of the "_mta-sts" TXT record, defined using ABNF[RFC7405], is as follows:

```
 sts-text-record  =  sts-version 1*(sts-field-delim sts-field)
                     [sts-field-delim]
 sts-field        =  sts-id /           ; Note that sts-id record
                     sts-extension      ; is required.
                …
```

- Diagram (in RFC 9085)



Human-written ⚠

Prone to ambiguities!

interoperability 😟 **correctness** 😟

Ambiguities can cause Security Vulnerabilities

Heartbleed Highlights a Contradiction in the Web

By Nicole Perlroth
April 18, 2014

Share full article

Heartbleed: Hundreds of thousands of servers at risk from catastrophic bug

Code error means that websites can leak user details including passwords through 'heartbeat' function used to secure connections

The Heartbleed bug: How a flaw in OpenSSL caused a security crisis

Analysis
Sep 6, 2022 • 10 mins

Internet | Open Source | Vulnerabilities

Ambiguity in **RFC 6520** causing bug in implementation of TLS heartbeat.

Ambiguities can cause Connection Failures

Ambiguity in **RFCs 9460 and 9461** causing incompatible behaviour in different DNS resolver implementations.

Existing work

No prior work has focused on **identifying logical ambiguities in RFCs.**

We present **the first framework** to address this problem.

Study

Manual classification of **273 verified technical errata reports** from Standards Track RFCs published between **January 2014 and January 2025**

Main Category (Total Count)	Sub-Category		Count
Inconsistency (202)	I-1	Direct inconsistency within or across specifications	119
	I-2	Indirect inconsistency within or across specifications	70
	I-3	Inconsistency with commonly accepted knowledge	13
Under-specification (37)	U-1	Direct under-specification due to undefined terms	7
	U-2	Direct under-specification due to incomplete constraints	15
	U-3	Indirect under-specification within or across specifications	10
	U-4	Under-specification due to incorrect or missing references	5
Others (34)		Editorial errors	15
		IANA considerations	13
		Suggestions or proposals	6

Study

Manual classification of **273 verified technical errata reports** from Standards Track RFCs published between **January 2014 and January 2025**

Main Category (Total Count)	Sub-Category		Count
Inconsistency (202)	I-1	Direct inconsistency within or across specifications	119
	I-2	Indirect inconsistency within or across specifications	70
	I-3	Inconsistency with commonly accepted knowledge	13
Under-specification (37)	U-1	Direct under-specification due to undefined terms	7
	U-2	Direct under-specification due to incomplete constraints	15
	U-3	Indirect under-specification within or across specifications	10
	U-4	Under-specification due to incorrect or missing references	5
Others (34)		Editorial errors	15
		IANA considerations	13
		Suggestions or proposals	6

Study

Manual classification of **273 verified technical errata reports** from Standards Track RFCs published between **January 2014 and January 2025**

Main Category (Total Count)	Sub-Category		Count
Inconsistency (202)	I-1	Direct inconsistency within or across specifications	119
	I-2	Ind	70
	I-3		
Under-specification (37)	U-1	D	
	U-2	Di	
	U-3	Inc	
	U-4	Ur	
Others (34)		Editorial	
		IANA co	
		Suggestions or proposals	6

RFC 7598
Section 4.3. S46 DMR Option
dmr-prefix6-len: Allowed values range from 0 to 128.

RFC 7599
Section 5.1. Destinations outside the MAP Domain
The DMR IPv6 prefix length SHOULD be 64 bits long by default and in any case MUST NOT exceed 96 bits

Is the maximum value 96 or 128? **Errata 4865**

Study

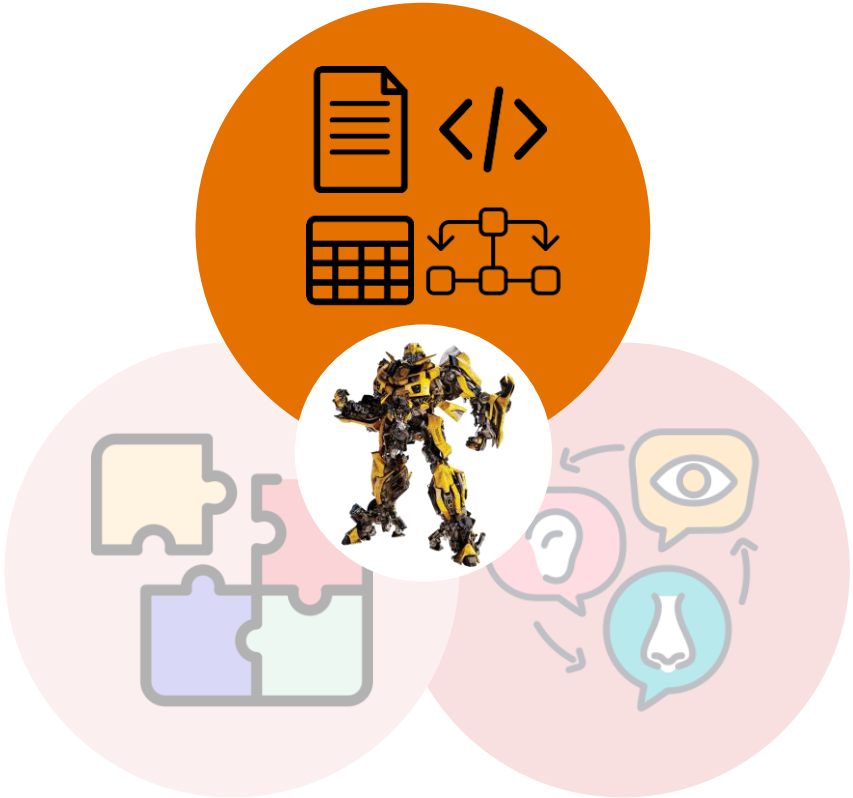
Manual classification of **273 verified technical errata reports** from Standards Track RFCs published betw

Main Category (Total Count)	Sub-Cate		
Inconsistency (202)	I-1	Di	
	I-2		
	I-3		
Under-specification (37)	U-1	Direct under-specification due to undefined terms	7
	U-2	Direct under-specification due to incomplete constraints	15
	U-3	Indirect under-specification within or across specifications	10
	U-4	Under-specification due to incorrect or missing references	5
Others (34)		Editorial errors	15
		IANA considerations	13
		Suggestions or proposals	6

RFC 8888
Section 3.1. RTCP Congestion Control Feedback Report
RTCP Congestion Control Feedback Packets SHOULD include a report - block for every active SSRC.

What is an *active* SSRC? **Errata 7894**

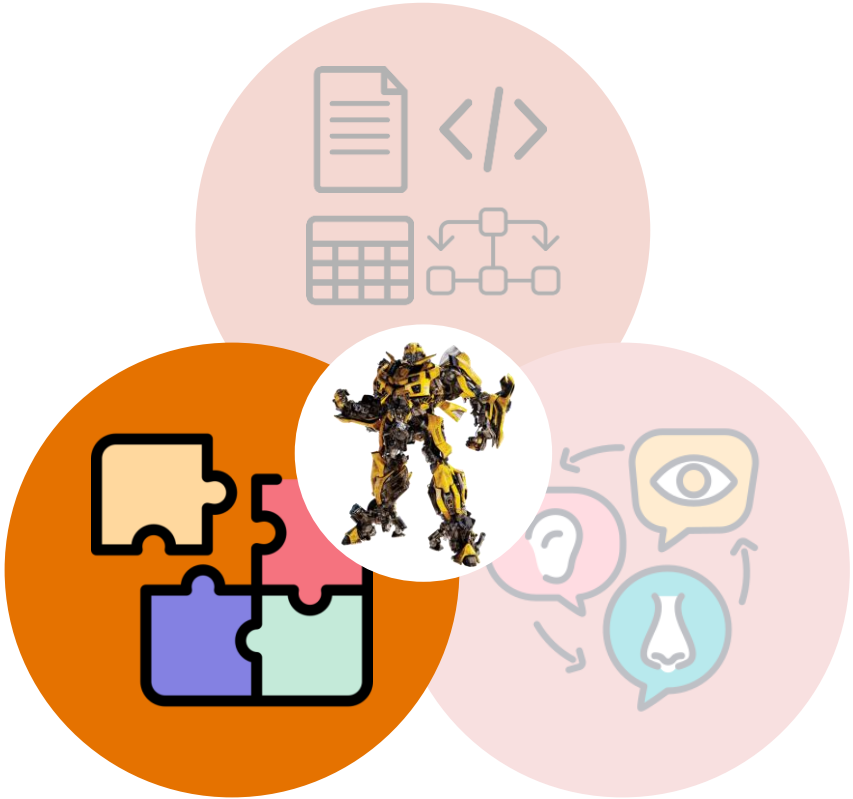
Why use LLMs to analyze RFCs?



Strength 1

Can process different kinds of formal and informal elements

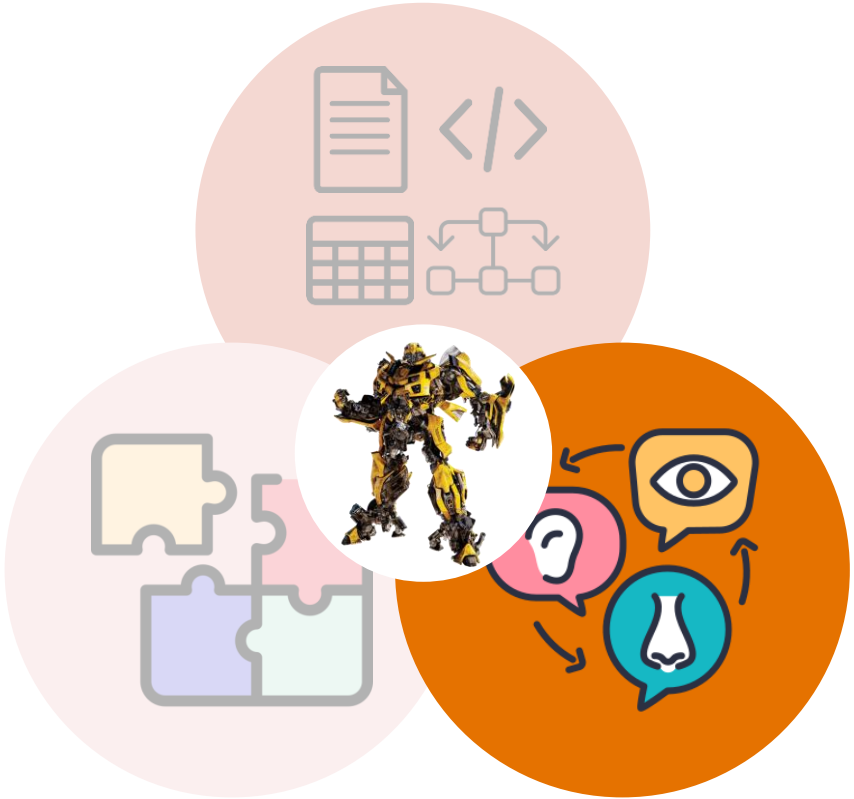
Why use LLMs to analyze RFCs?



Strength 1
Can process different kinds of formal and informal elements

Strength 2
Can perform complex reasoning

Why use LLMs to analyze RFCs?



Strength 1
Can process different kinds of formal and informal elements

Strength 2
Can perform complex reasoning

Strength 3
Possess common technical knowledge

Challenges with LLMs to detect ambiguities in RFCs

- **Long specification documents**
- Multi-document reasoning
- Limited domain knowledge
- Prone to hallucination

RFC 1035 which describes DNS is **55 pages** long!

Challenges with LLMs to detect ambiguities in RFCs

- Long specification documents
- **Multi-document reasoning**
- Limited domain knowledge
- Prone to hallucination

RFC 9460

9.6. Use of HTTPS RRs in Other Protocols

All HTTP connections to named origins are eligible to use HTTPS RRs, even when HTTP is used as part of another protocol or without an explicit HTTP-related URI scheme (Section 4.2 of [HTTP]). For example, clients that support HTTPS RRs and implement [WebSocket] using the altered opening handshake from [FETCH-WEBSOCKETS] SHOULD use HTTPS RRs for the requestURL.

RFC 9110

RFC 6455

WebSockets Living Standard

Challenges with LLMs to detect ambiguities in RFCs

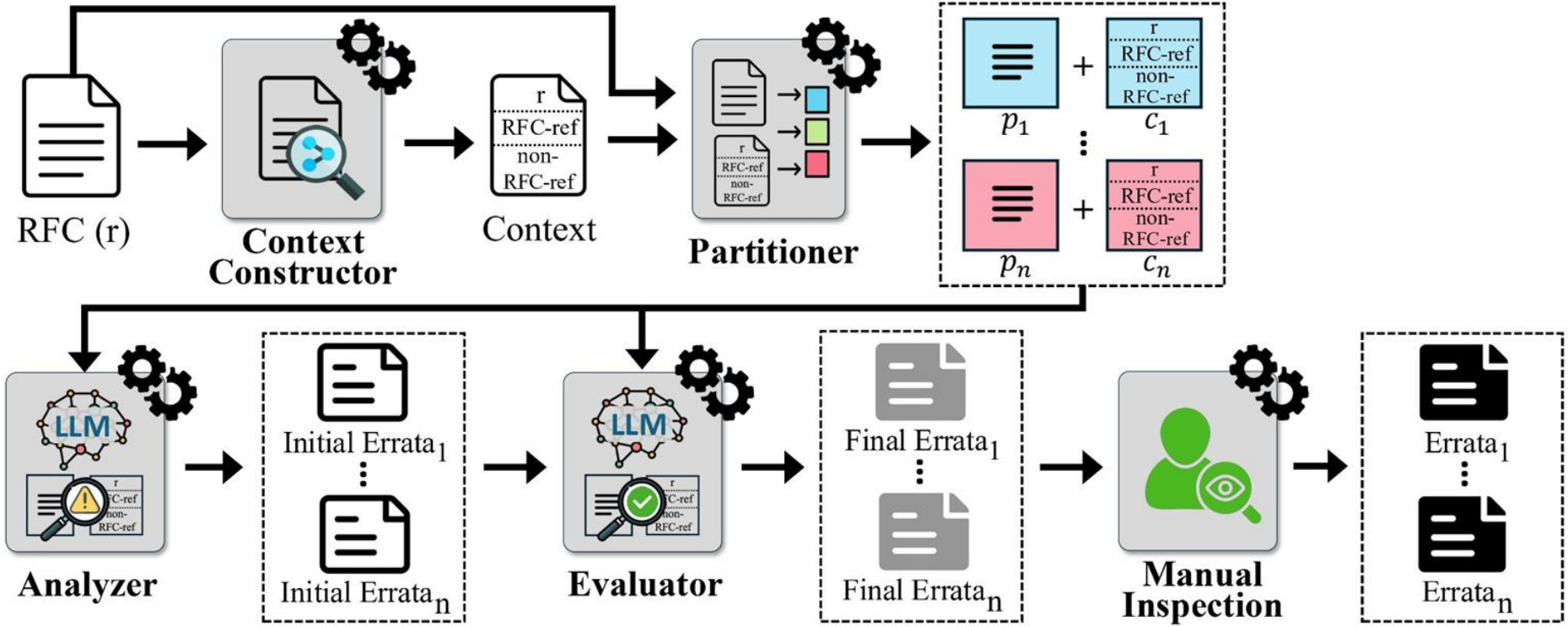
- Long specification documents
- Multi-document reasoning
- **Limited domain knowledge**
- Prone to hallucination

LLMs do not *know* what kind of ambiguities to look for and the strategies to find them.

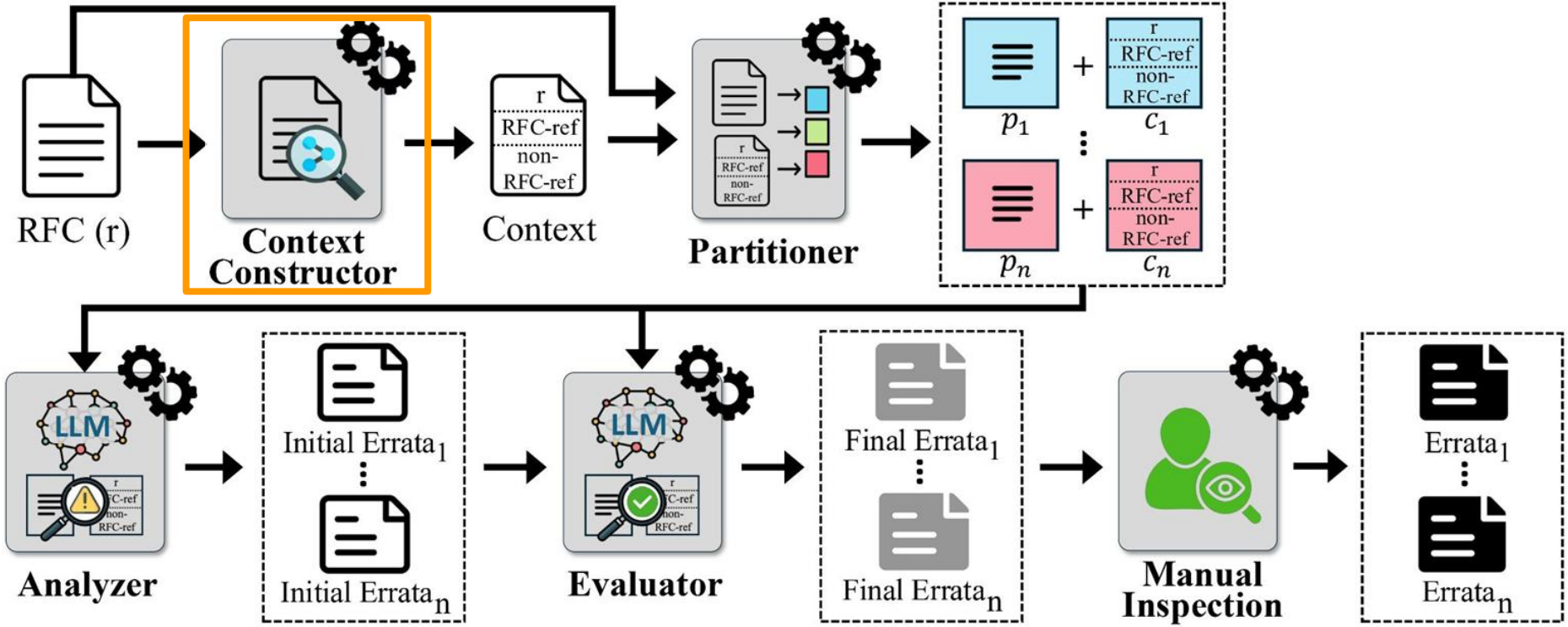
Challenges with LLMs to detect ambiguities in RFCs

- Long specification documents
- Multi-document reasoning
- Limited domain knowledge
- **Prone to hallucination**

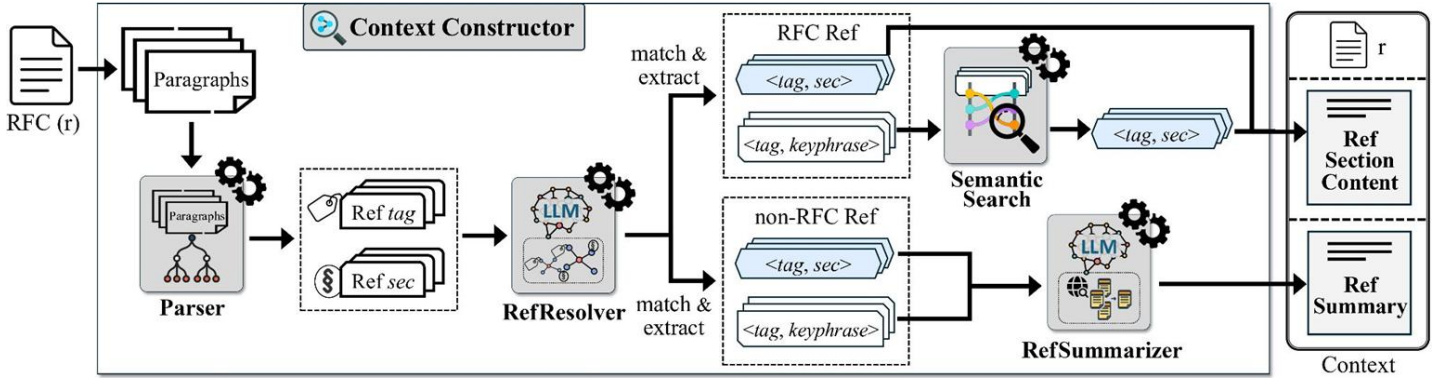
RFCScope — Overview



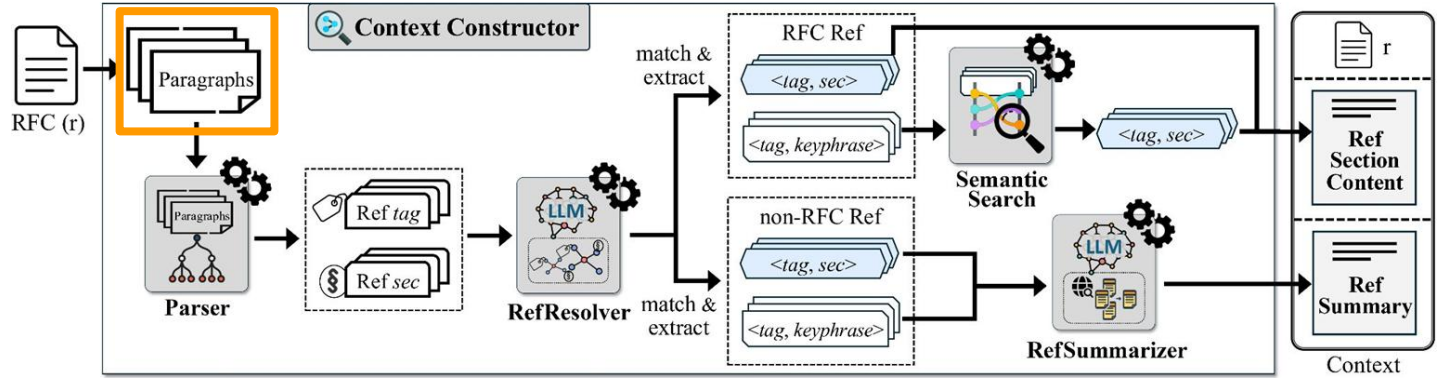
RFCScope — Overview



RFCScope — Context Constructor



RFCScope — Context Constructor

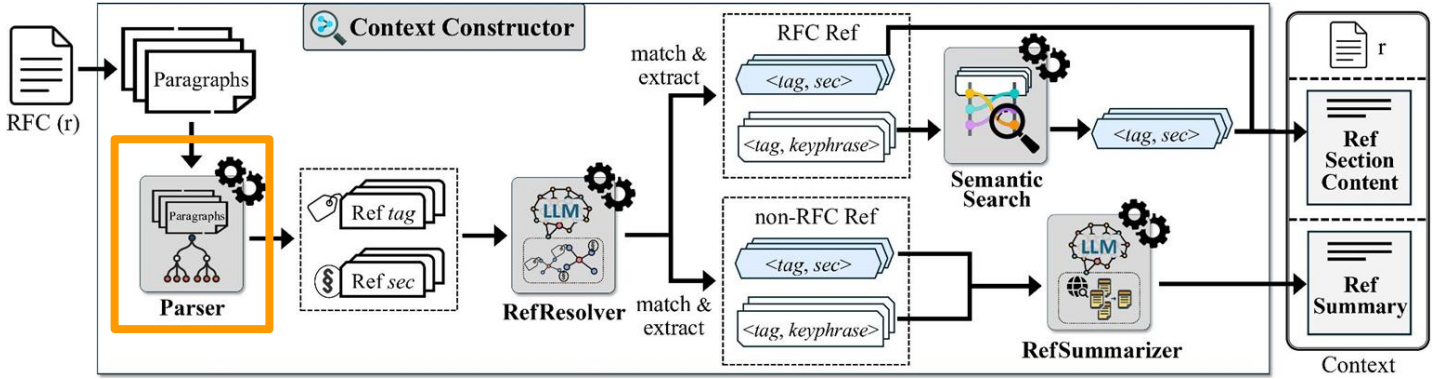


RFC 9460

9.6. Use of HTTPS RRs in Other Protocols

All HTTP connections to named origins are eligible to use HTTPS RRs, even when HTTP is used as part of another protocol or without an explicit HTTP-related URI scheme (Section 4.2 of [HTTP]). For example, clients that support HTTPS RRs and implement [WebSocket] using the altered opening handshake from [FETCH-WEBSOCKETS] SHOULD use HTTPS RRs for the requestURL.

RFCScope — Context Constructor

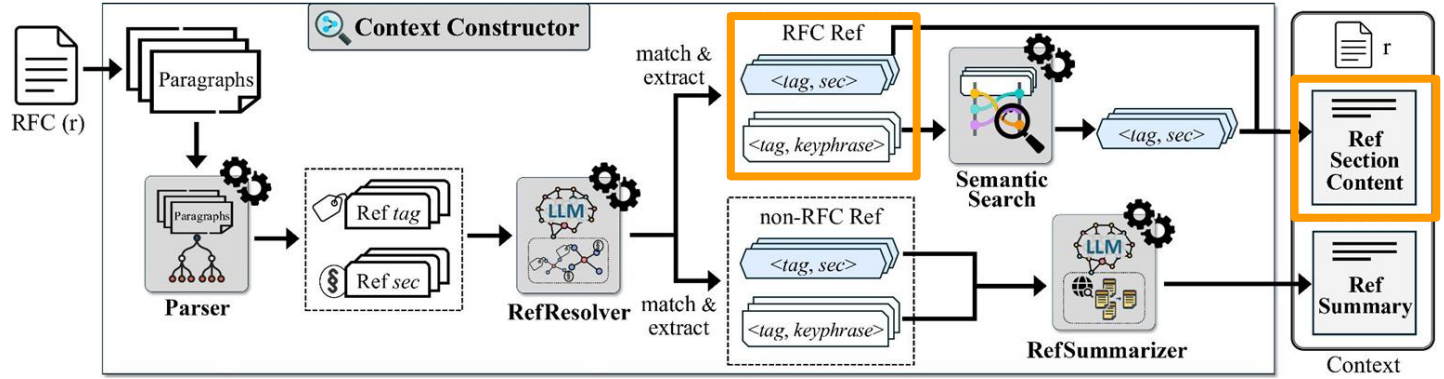


RFC 9460

9.6. Use of HTTPS RRs in Other Protocols

All HTTP connections to named origins are eligible to use HTTPS RRs, even when HTTP is used as part of another protocol or without an explicit HTTP-related URI scheme (Section 4.2 of [HTTP]). For example, clients that support HTTPS RRs and implement [WebSocket] using the altered opening handshake from [FETCH-WEBSOCKETS] SHOULD use HTTPS RRs for the requestURL.

RFCScope — Context Constructor



RFC 9460

9.6. Use of HTTPS RRs in Other Protocols

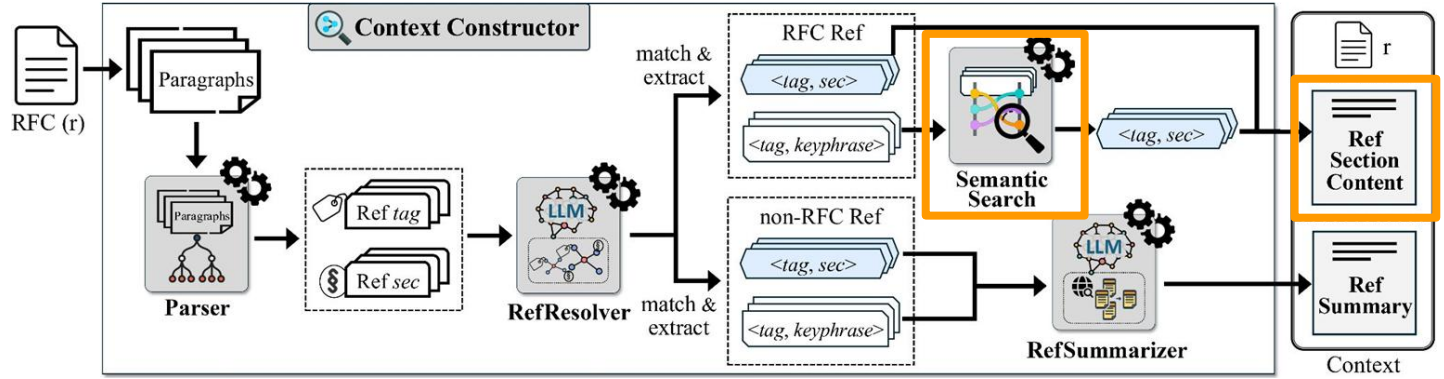
All HTTP connections to named origins are eligible to use HTTPS RRs, even when HTTP is used as part of another protocol or without an explicit HTTP-related URI scheme (Section 4.2 of [HTTP]). For example, clients that support HTTPS RRs and implement [WebSocket] using the altered opening handshake from [FETCH-WEBSOCKETS] SHOULD use HTTPS RRs for the requestURL.

Section 4.2

RFC 9110

RFC 9110
Section 4.2

RFCScope — Context Constructor



RFC 9460

9.6. Use of HTTPS RRs in Other Protocols

All HTTP connections to named origins are eligible to use HTTPS RRs, even when HTTP is used as part of another protocol or without an explicit HTTP-related URI scheme (Section 4.2 of [HTTP]). For example, clients that support HTTPS RRs and implement [WebSocket] using the altered opening handshake from [FETCH-WEBSOCKETS] SHOULD use HTTPS RRs for the requestURL.

RFC 9110
Section 4.2

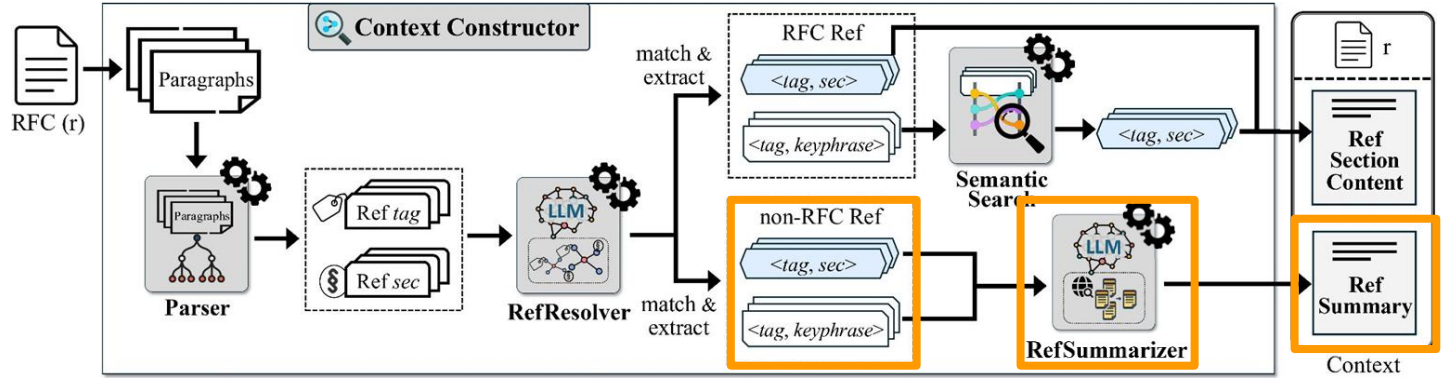
"opening handshake"
RFC 6455

RFC 6455
Section 4

RFC 6455
Section 3.1

RFC 6455
Section 5.1

RFCScope — Context Constructor



RFC 9460

9.6. Use of HTTPS RRs in Other Protocols

All HTTP connections to named origins are eligible to use HTTPS RRs, even when HTTP is used as part of another protocol or without an explicit HTTP-related URI scheme (Section 4.2 of [HTTP]). For example, clients that support HTTPS RRs and implement [WebSocket] using the altered opening handshake from [FETCH-WEBSOCKETS] SHOULD use HTTPS RRs for the requestURL.

RFC 9110 Section 4.2

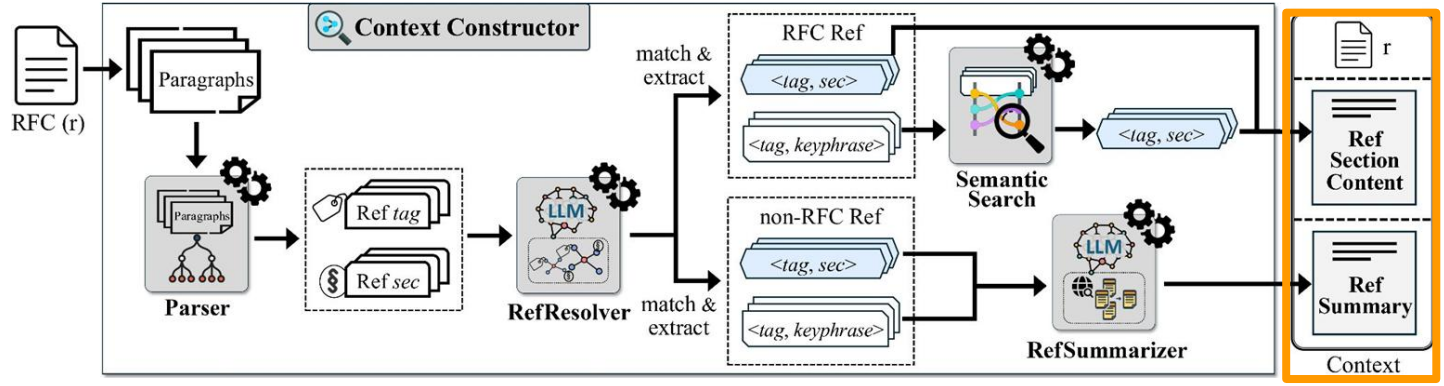
RFC 6455 Section 4

"opening handshake"

websockets.spec.whatwg.org

WebSockets Living Standard
Summary of "opening handshake"

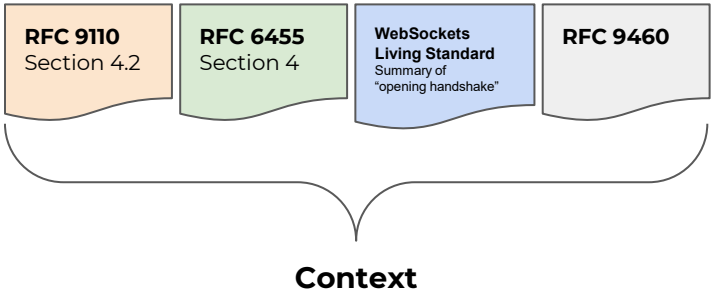
RFCScope — Context Constructor



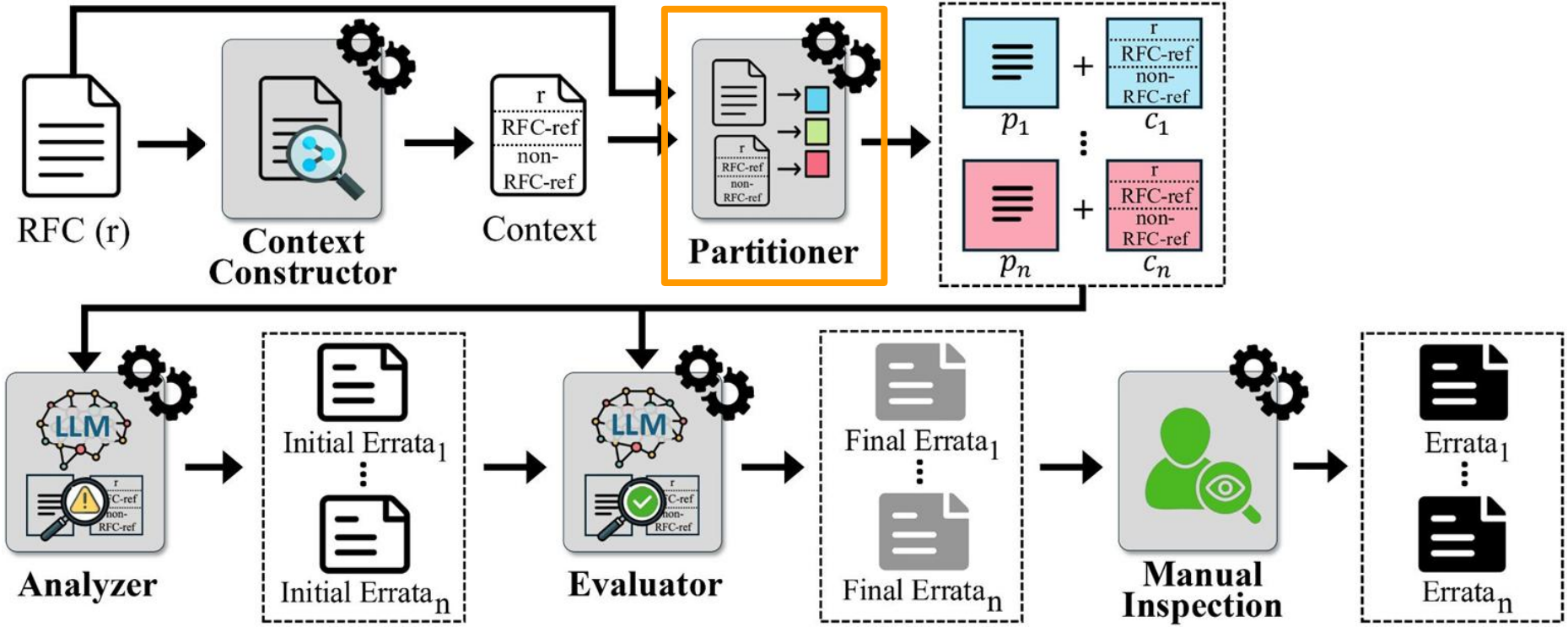
RFC 9460

9.6. Use of HTTPS RRs in Other Protocols

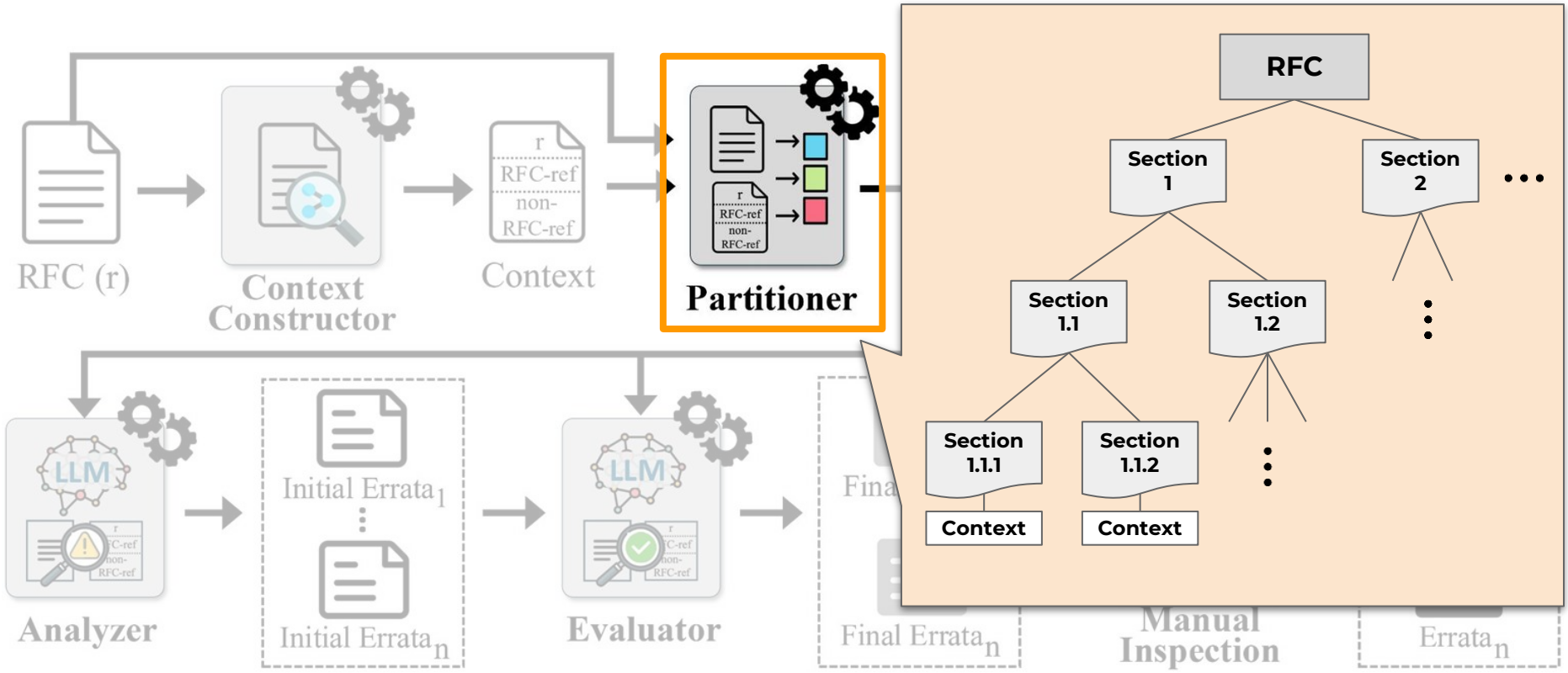
All HTTP connections to named origins are eligible to use HTTPS RRs, even when HTTP is used as part of another protocol or without an explicit HTTP-related URI scheme (Section 4.2 of [HTTP]). For example, clients that support HTTPS RRs and implement [WebSocket] using the altered opening handshake from [FETCH-WEBSOCKETS] SHOULD use HTTPS RRs for the requestURL.



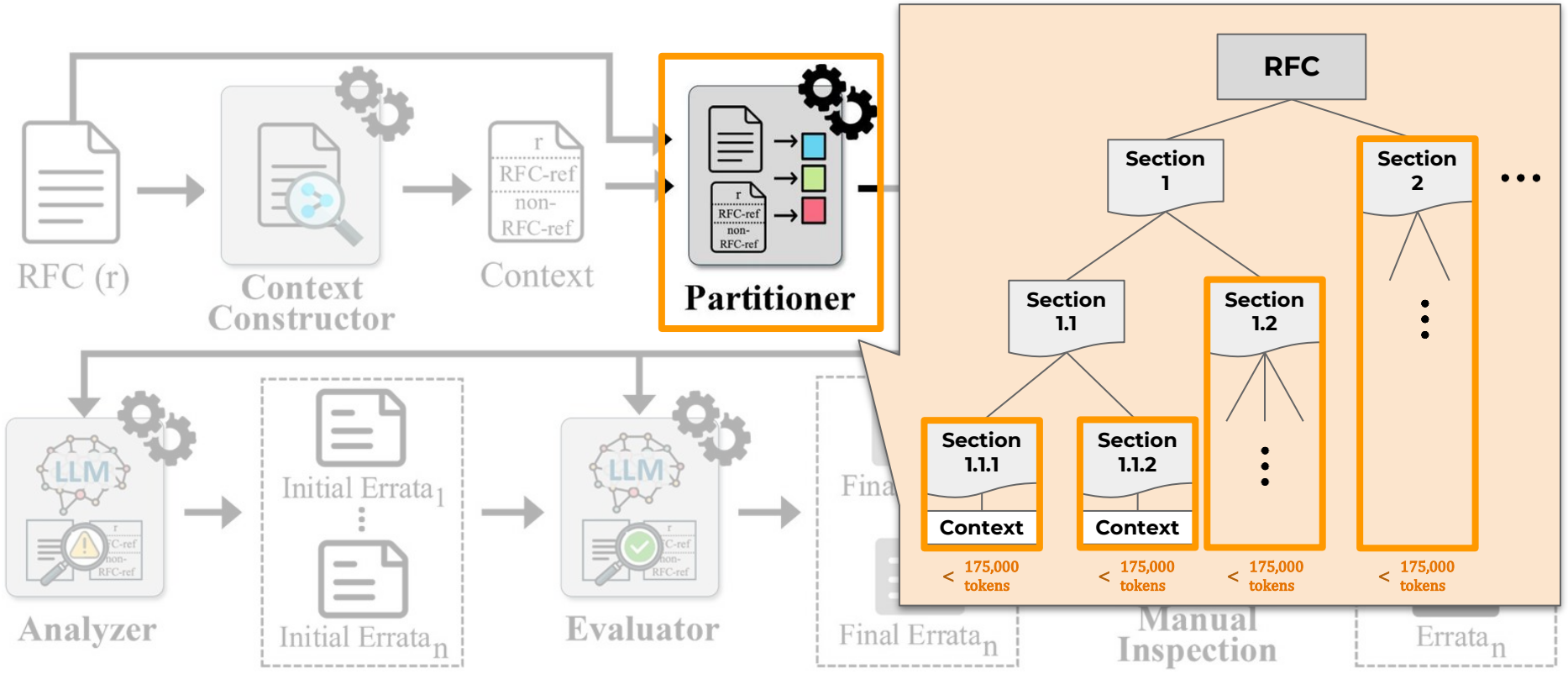
RFCScope — Overview



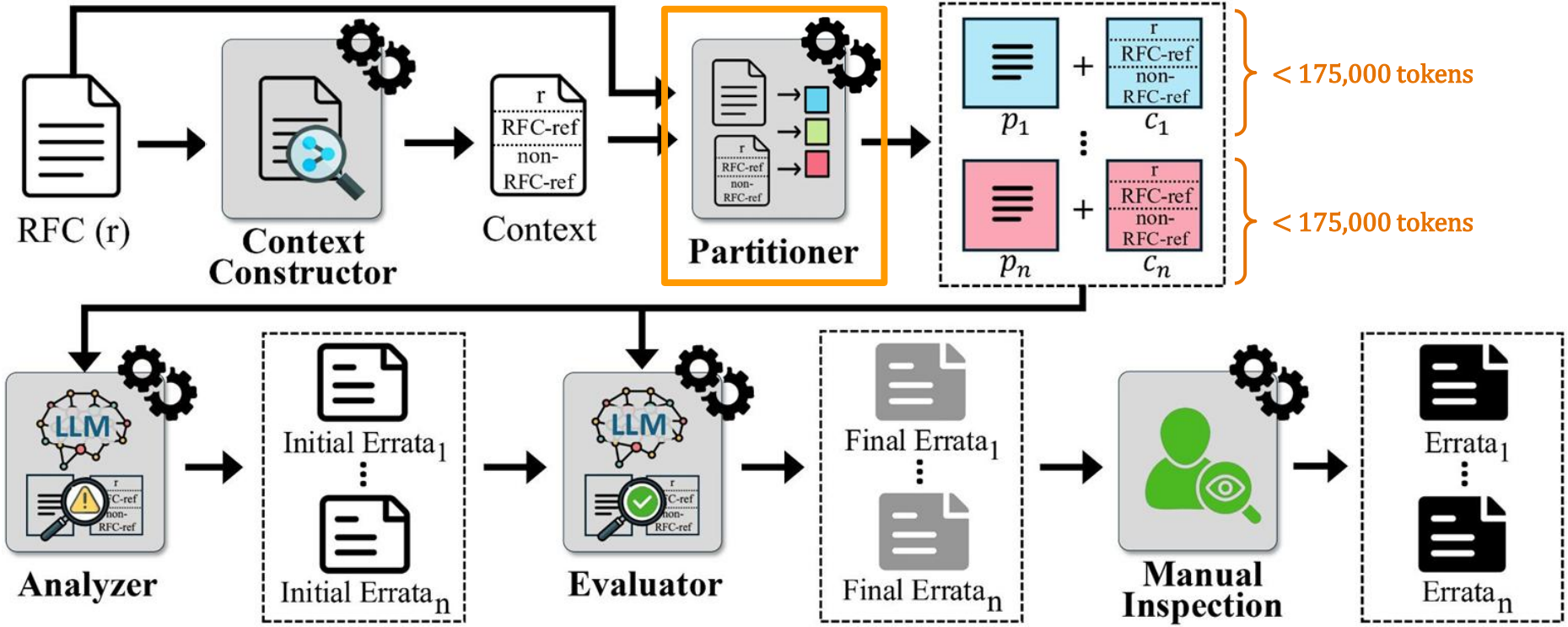
RFCScope — Overview



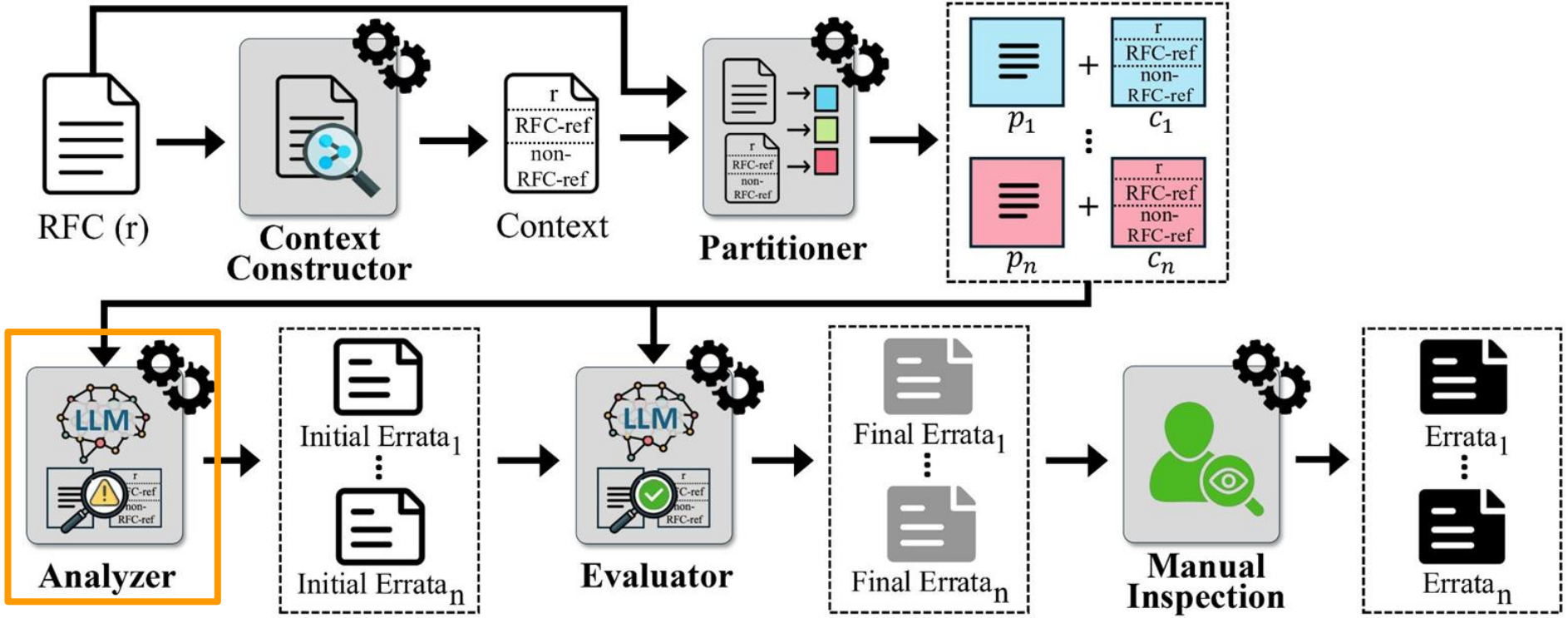
RFCScope — Overview



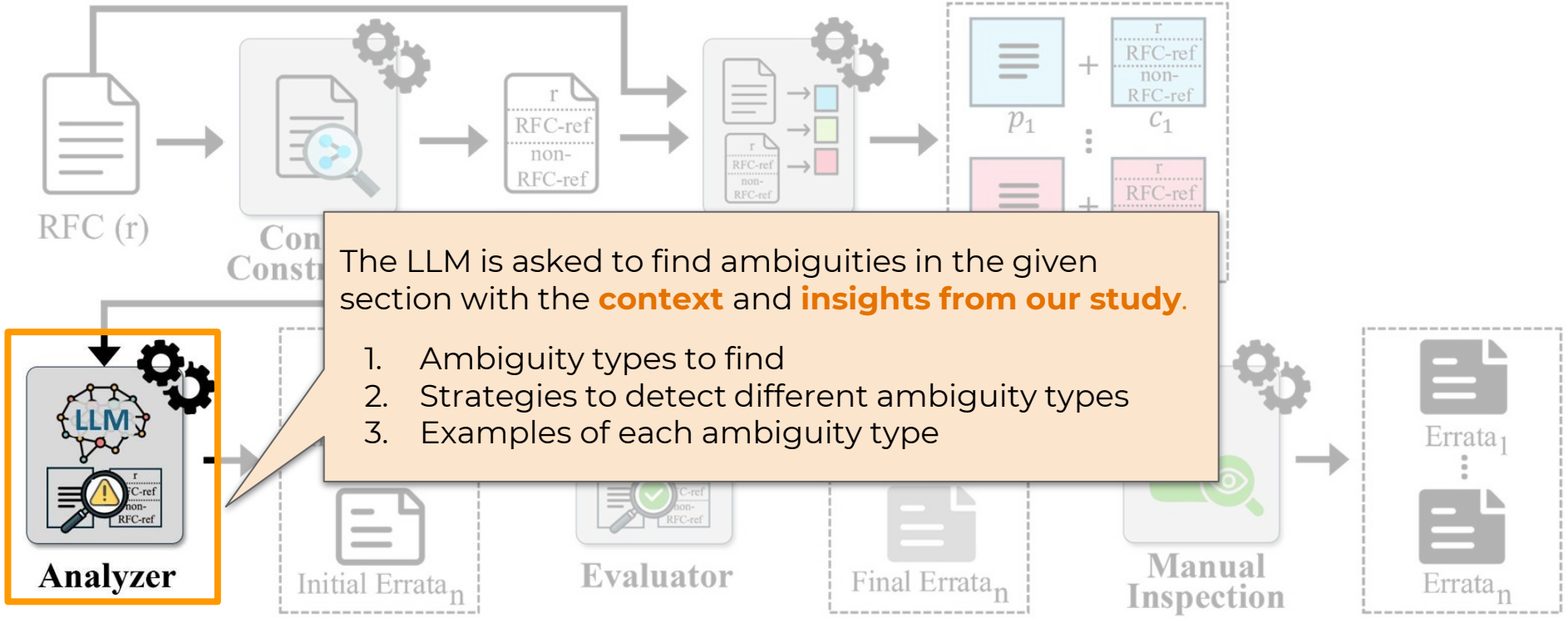
RFCScope — Overview



RFCScope — Overview



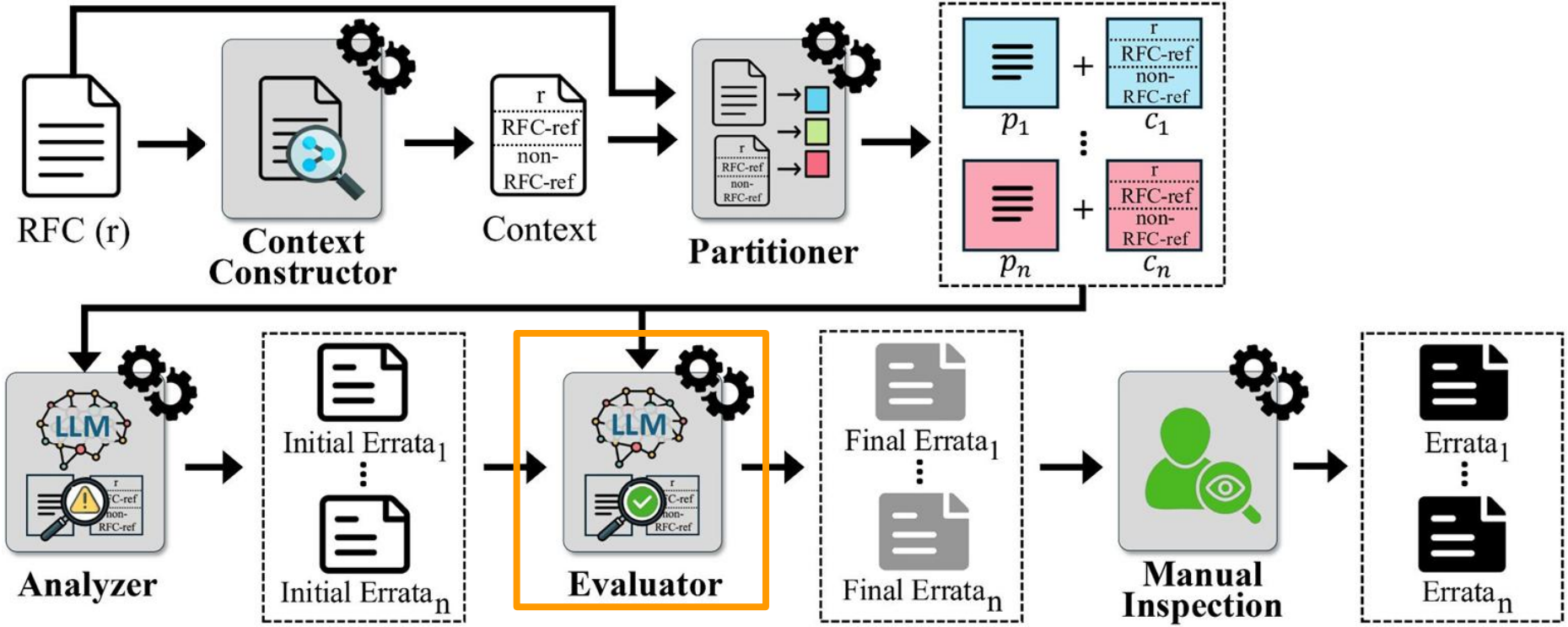
RFCScope — Overview



The LLM is asked to find ambiguities in the given section with the **context** and **insights from our study**.

1. Ambiguity types to find
2. Strategies to detect different ambiguity types
3. Examples of each ambiguity type

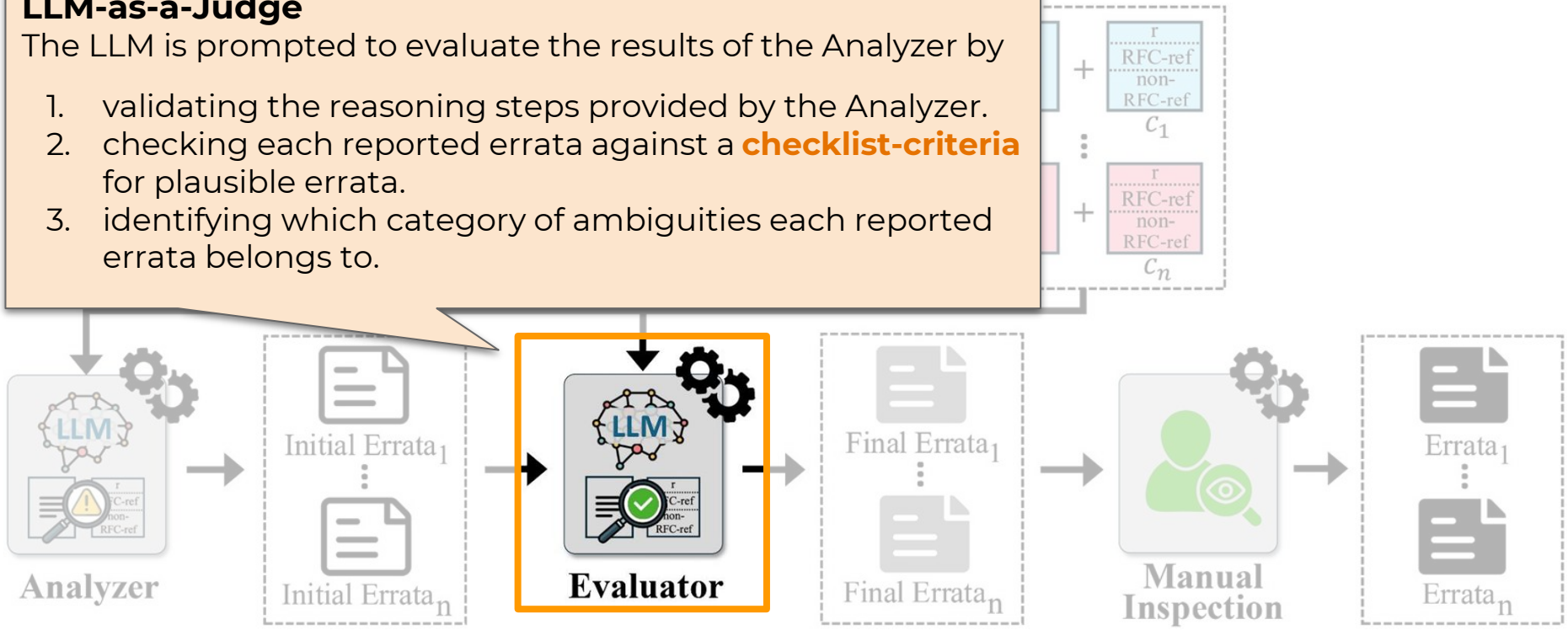
RFCScope — Overview



RFCScope — Overview

LLM-as-a-Judge
The LLM is prompted to evaluate the results of the Analyzer by

1. validating the reasoning steps provided by the Analyzer.
2. checking each reported errata against a **checklist-criteria** for plausible errata.
3. identifying which category of ambiguities each reported errata belongs to.

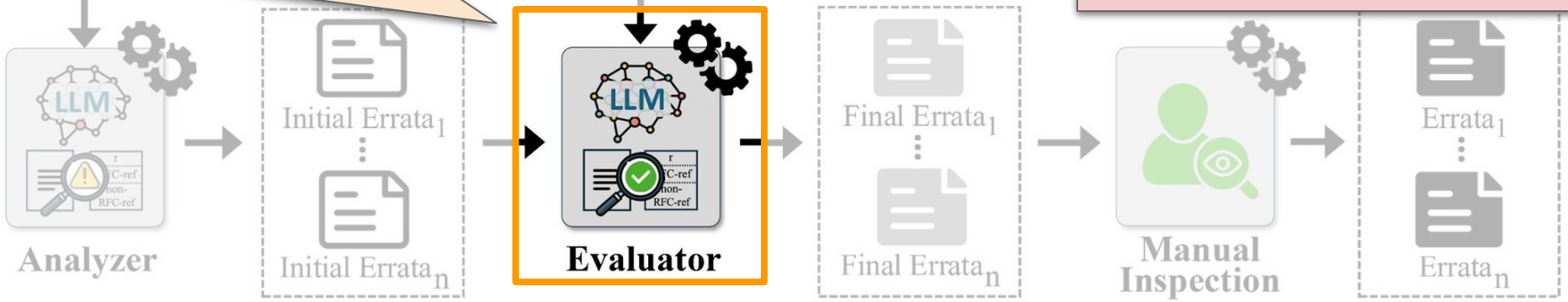


RFCScope — Overview

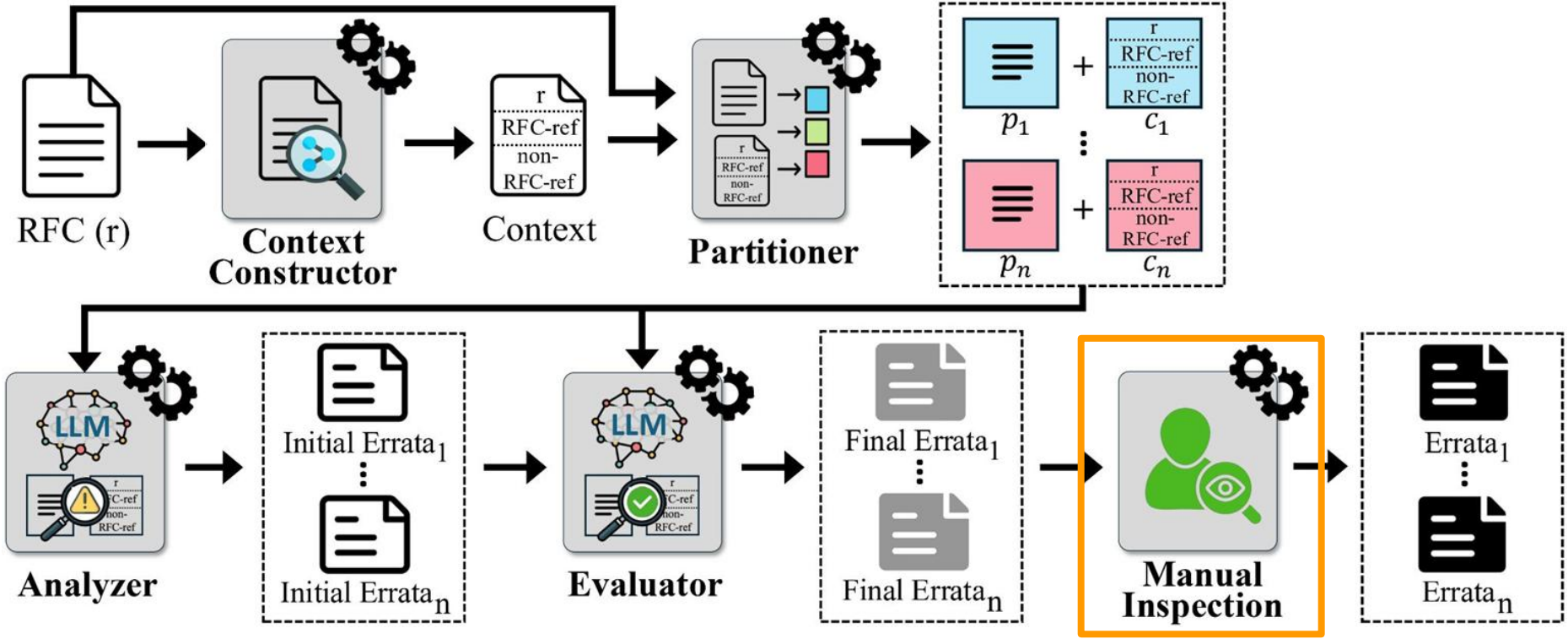
LLM-as-a-Judge
The LLM is prompted to evaluate the results of the Analyzer by

1. validating the reasoning steps provided by the Analyzer.
2. checking each reported errata against a **checklist-criteria** for plausible errata.
3. identifying which category of ambiguities each reported errata belongs to.

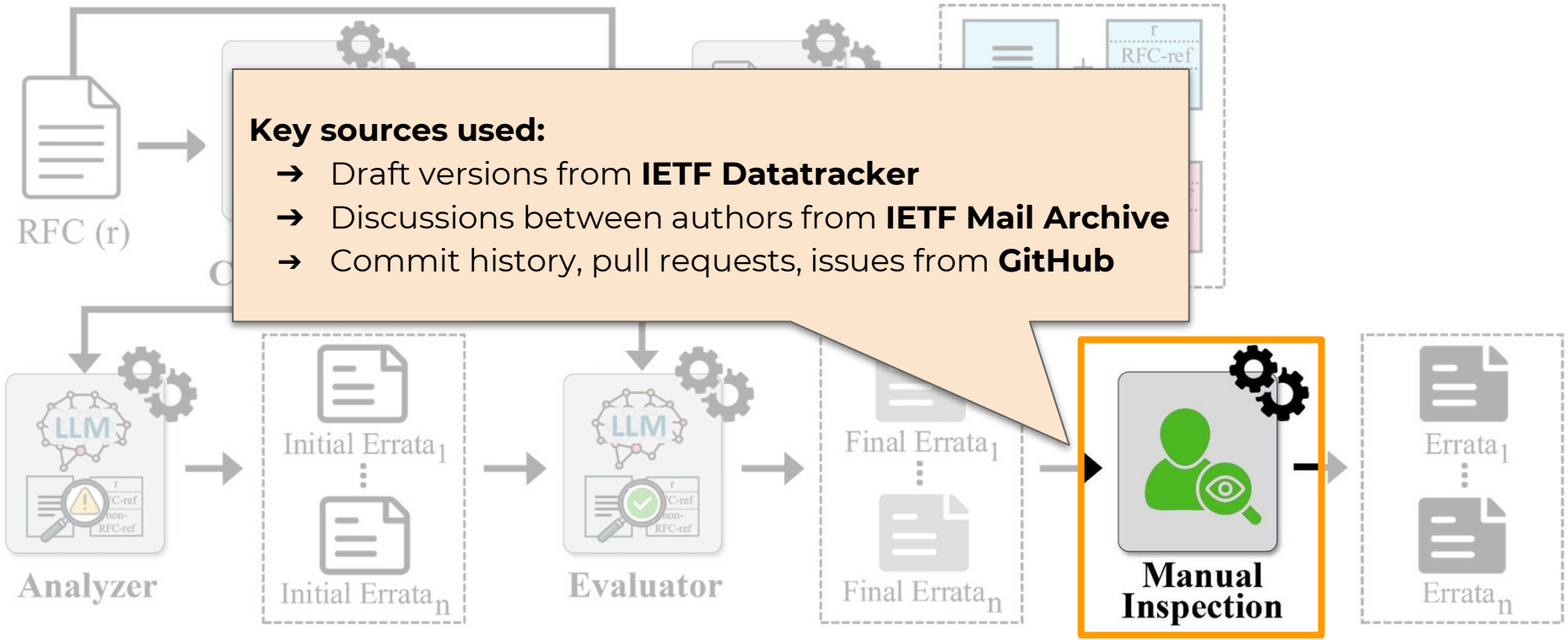
- Is this information given later in the document?
- Is this an implementation detail left to the discretion of the user?
- Is clarification or elaboration relevant to this document?



RFCScope — Overview



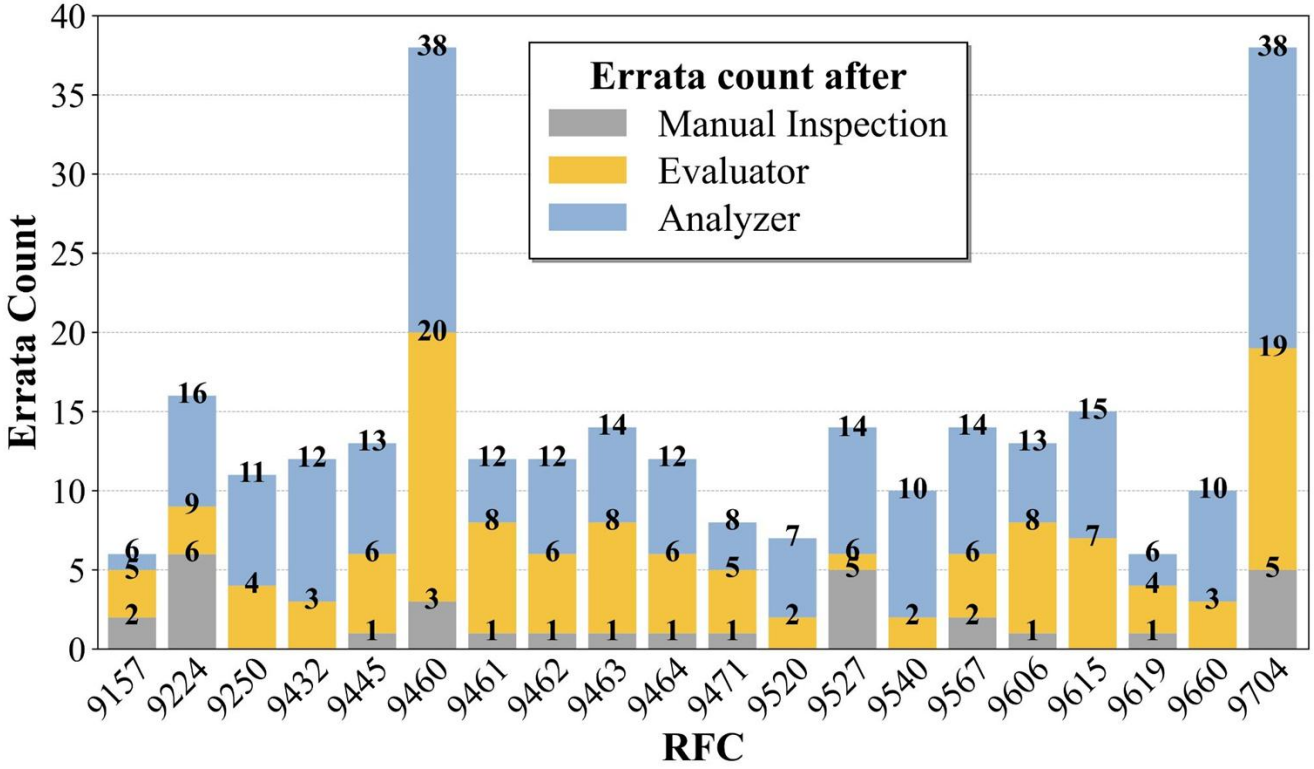
RFCScope — Overview



Key sources used:

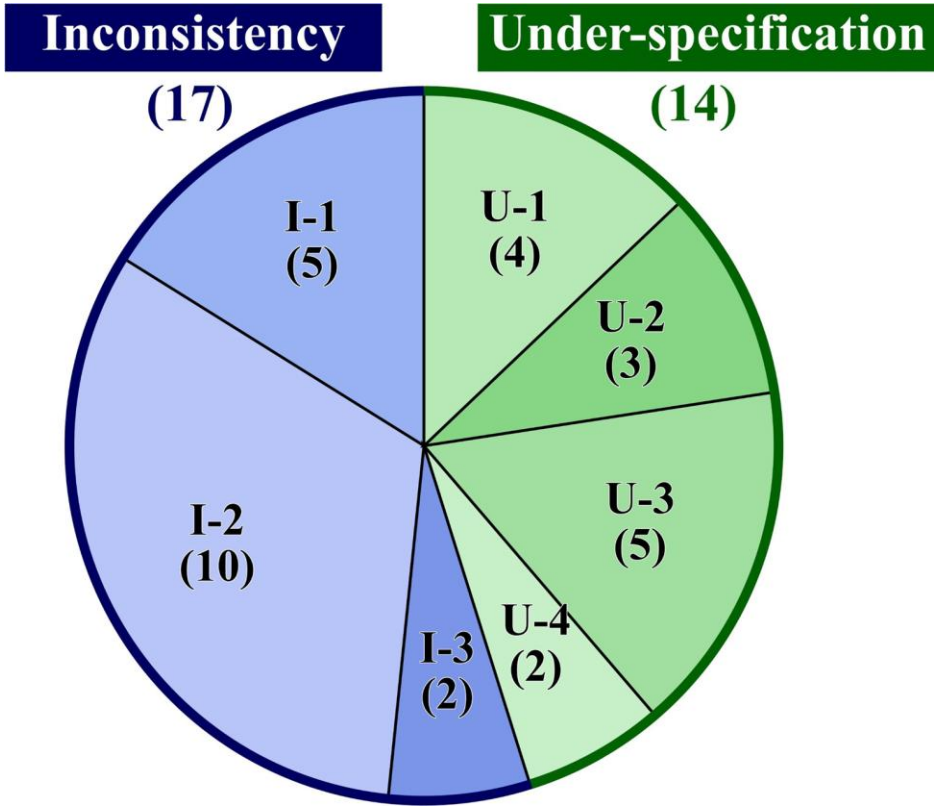
- ➔ Draft versions from **IETF Datatracker**
- ➔ Discussions between authors from **IETF Mail Archive**
- ➔ Commit history, pull requests, issues from **GitHub**

Evaluation

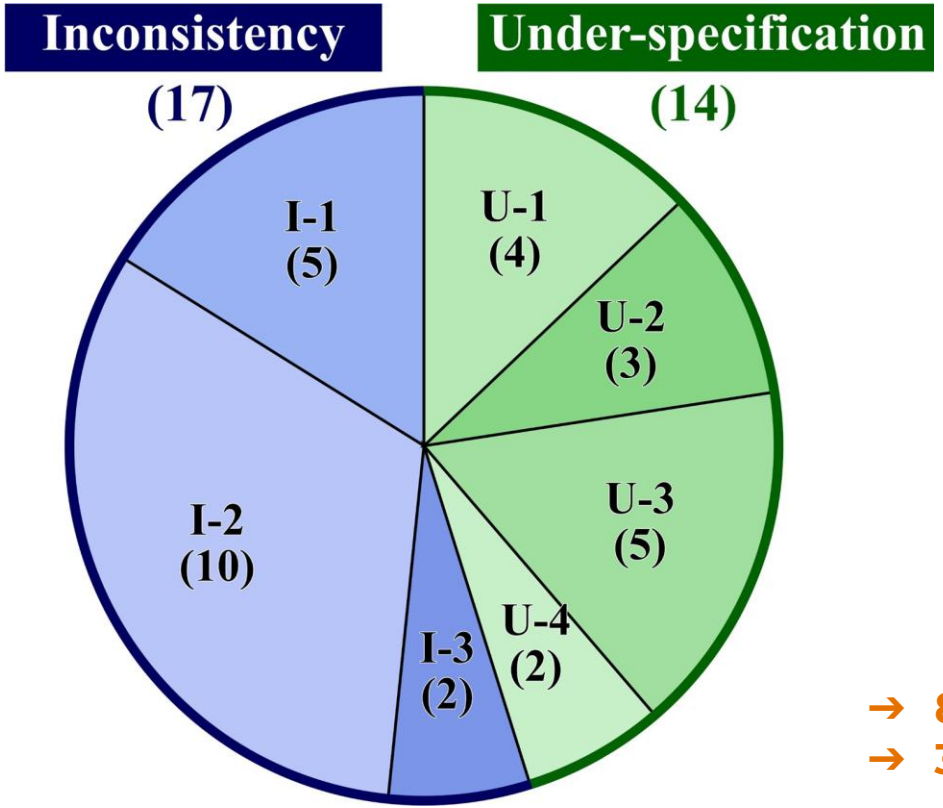


20 recent DNS-related RFCs

Evaluation

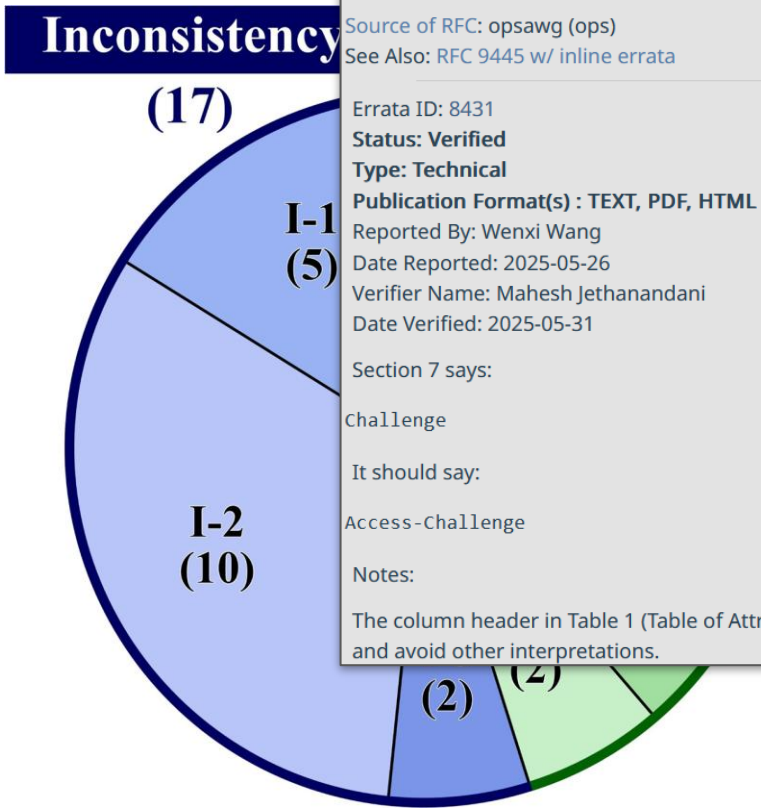

Inconsistency
(17)
Under-specification
(14)
I-1
(5)
U-1
(4)
U-2
(3)
U-3
(5)
U-4
(2)
I-3
(2)
I-2
(10)

Evaluation



→ **8 errata confirmed** by RFC authors
→ **3 officially verified** technical errata

Evaluation



RFC 9445, "RADIUS Extensions for DHCP-Configured Services", August 2023

Source of RFC: opsawg (ops)
See Also: RFC 9445 w/ inline errata

Errata ID: 8431
Status: Verified
Type: Technical
Publication Format(s) : TEXT, PDF, HTML
Reported By: Wenxi Wang
Date Reported: 2025-05-26
Verifier Name: Mahesh Jethanandani
Date Verified: 2025-05-31

Section 7 says:

```
Challenge
```

It should say:

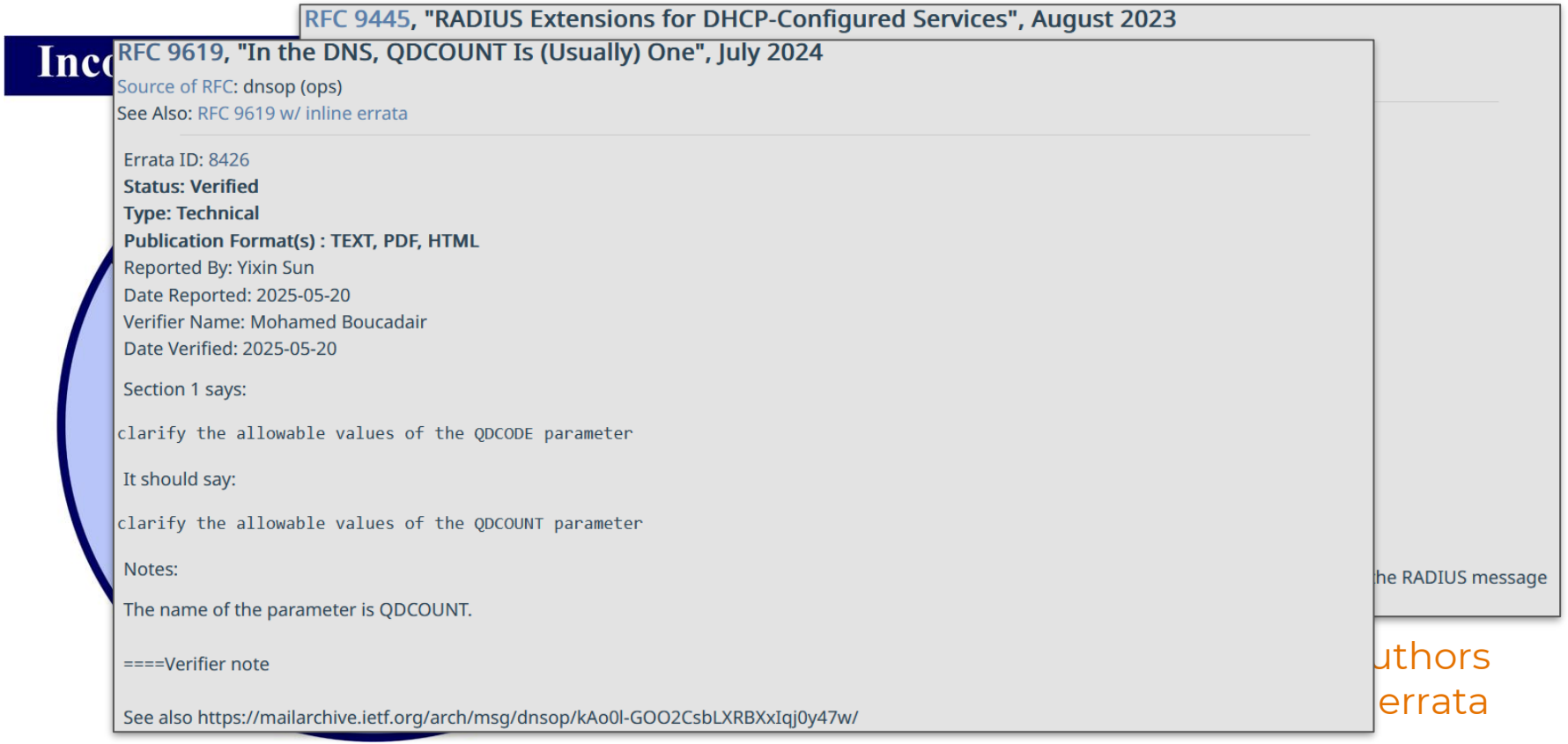
```
Access-Challenge
```

Notes:

The column header in Table 1 (Table of Attributes) says "Challenge", which should be Access-Challenge to reflect the name of the RADIUS message and avoid other interpretations.

→ **8 errata confirmed** by RFC authors
→ **3 officially verified** technical errata

Evaluation


RFC 9445, "RADIUS Extensions for DHCP-Configured Services", August 2023
RFC 9619, "In the DNS, QDCOUNT Is (Usually) One", July 2024
Source of RFC: dnsop (ops)
See Also: RFC 9619 w/ inline errata
Errata ID: 8426
Status: Verified
Type: Technical
Publication Format(s) : TEXT, PDF, HTML
Reported By: Yixin Sun
Date Reported: 2025-05-20
Verifier Name: Mohamed Boucadair
Date Verified: 2025-05-20
Section 1 says:
clarify the allowable values of the QDCODE parameter
It should say:
clarify the allowable values of the QDCOUNT parameter
Notes:
The name of the parameter is QDCOUNT.
====Verifier note
See also https://mailarchive.ietf.org/arch/msg/dnsop/kAo0l-GOO2CsbLXRBXxIqj0y47w/
he RADIUS message
uthors
errata

Evaluation

RFC 9445, "RADIUS Extensions for DHCP-Configured Services", August 2023

RFC 9619, "In the DNS, QDCOUNT Is (Usually) One", July 2024

RFC 9704, "Establishing Local DNS Authority in Validated Split-Horizon Environments", January 2025

Source of RFC: add (int)
See Also: RFC 9704 w/ inline errata

Errata ID: 8590
Status: Verified
Type: Technical
Publication Format(s) : TEXT, PDF, HTML
Reported By: Mrigank Pawagi
Date Reported: 2025-10-01
Verifier Name: Éric Vyncke
Date Verified: 2025-10-02

Section 6.2 says:

```
performs full DNSSEC validation locally [RFC6698]
```

It should say:

```
performs full DNSSEC validation locally [RFC4033]
```

Notes:

Wrong RFC reference. The correct reference is RFC 4033 for DNSSEC and "Validating Stub Resolver" definition.

--- Verifier note (Eric Vyncke) ----
Thanks for Dan Wing for the verification.

he RADIUS message

uthors
errata

Contributions

★ **Study**

First study of technical errata from RFCs to propose a **taxonomy of logical ambiguities.**

★ **Framework**

First framework for detecting logical ambiguities in RFCs.

★ **Real-world impact**

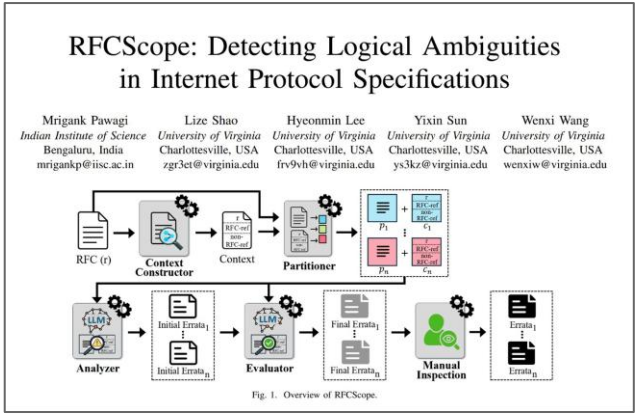
31 new logical ambiguities across 14 recent DNS-related RFCs, with 8 confirmed by RFC authors and 3 officially verified as technical errata.

Learn more about RFCScope

Try RFCScope

or visit
github.com/HIPREL-Group/RFCScope

Check out our paper


RFCScope: Detecting Logical Ambiguities in Internet Protocol Specifications

Mrigank Pawagi, Indian Institute of Science, Bengaluru, India, mrigankp@iisc.ac.in
Lize Shao, University of Virginia, Charlottesville, USA, zgr3et@virginia.edu
Hyeonmin Lee, University of Virginia, Charlottesville, USA, frv9vh@virginia.edu
Yixin Sun, University of Virginia, Charlottesville, USA, ys3kz@virginia.edu
Wenxi Wang, University of Virginia, Charlottesville, USA, wenxiw@virginia.edu

Fig. 1. Overview of RFCScope.

and come by our poster!