Infinite data. One answer.

Exploiting Reconfigurability for Text Search

Roger D. Chamberlain, Mark A. Franklin, and Ron S. Indeck

Exegy Inc.

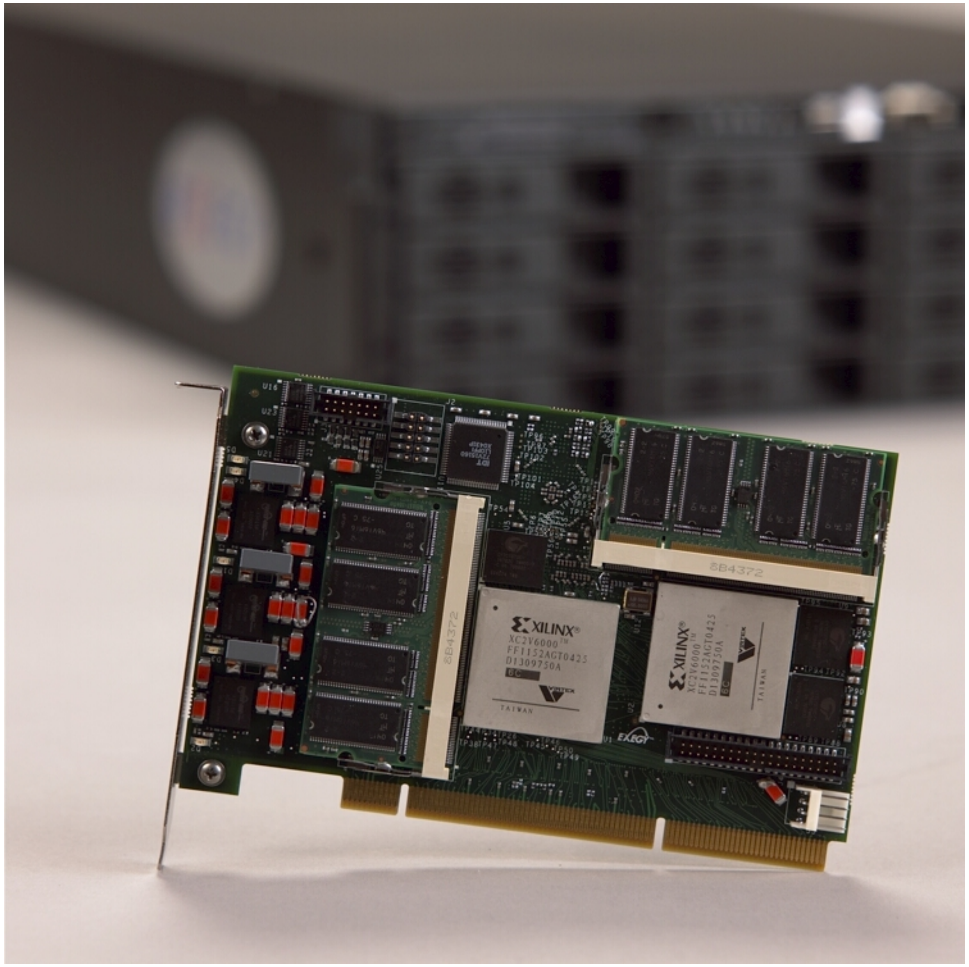
Exegy TextMiner

Highly Optimized Data Pipeline from Input thru Output

Specialized Processing in Close Proximity to Data

1-7 TB fast RAID; RAM / FPGA contiguous

Specialized Processing on Custom Board

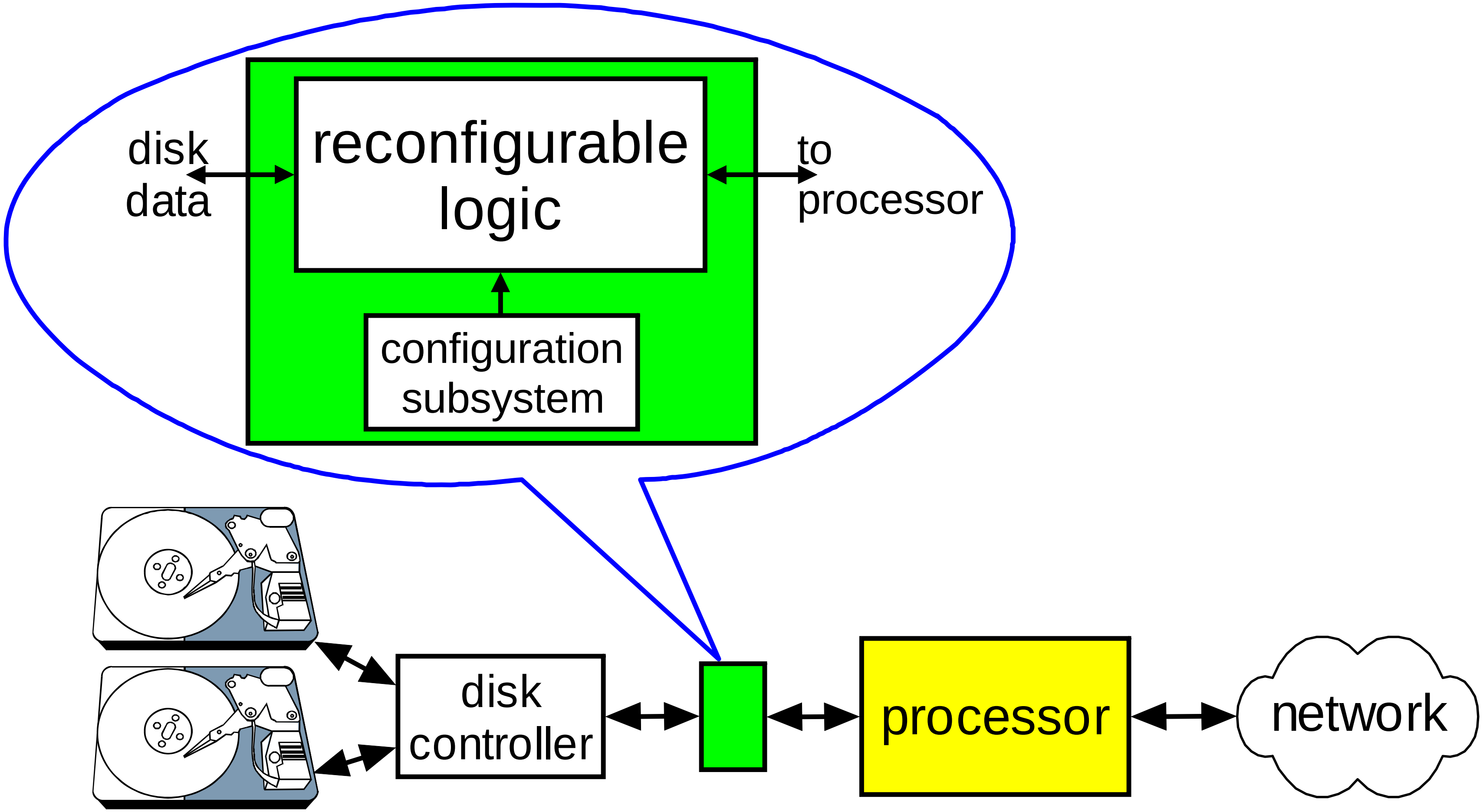


FPGA accelerated custom board

- Permits massively parallel operations
- Offloads work from CPU
- Integrates with other system components enabling high-speed data ingress and egress
- Designed with common APIs to give user control of functionality
- Draws from a library of pre-defined modules used to perform certain operations
- New functional modules readily incorporated

Analogous to graphic accelerator cards

Exegy A2000 Appliance



TextMiner Application

- Searching through an unindexed text corpus for items of interest
- Example query

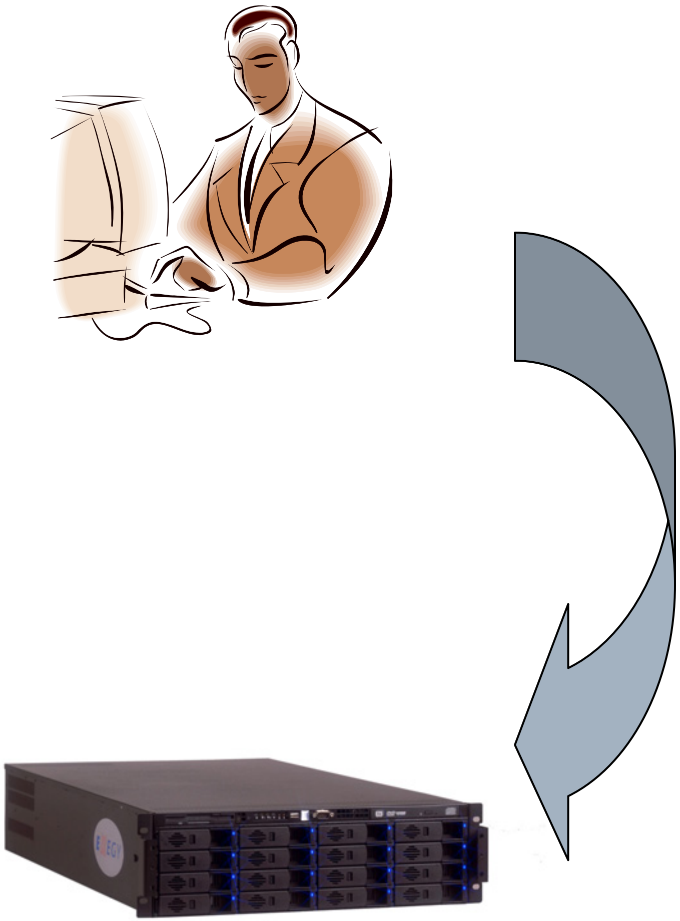
  (Cardinals NEAR[200] Baseball) AND
  (Manchester NEAR[200] Soccer)

  - “Cardinals” within 200 characters of “Baseball” and
  - “Manchester” within 200 characters of “Soccer”
- Supported combining operators include
  - Boolean: AND, OR, NOT
  - Proximity: NEAR, ANDTHEN

Benefits of reconfiguration

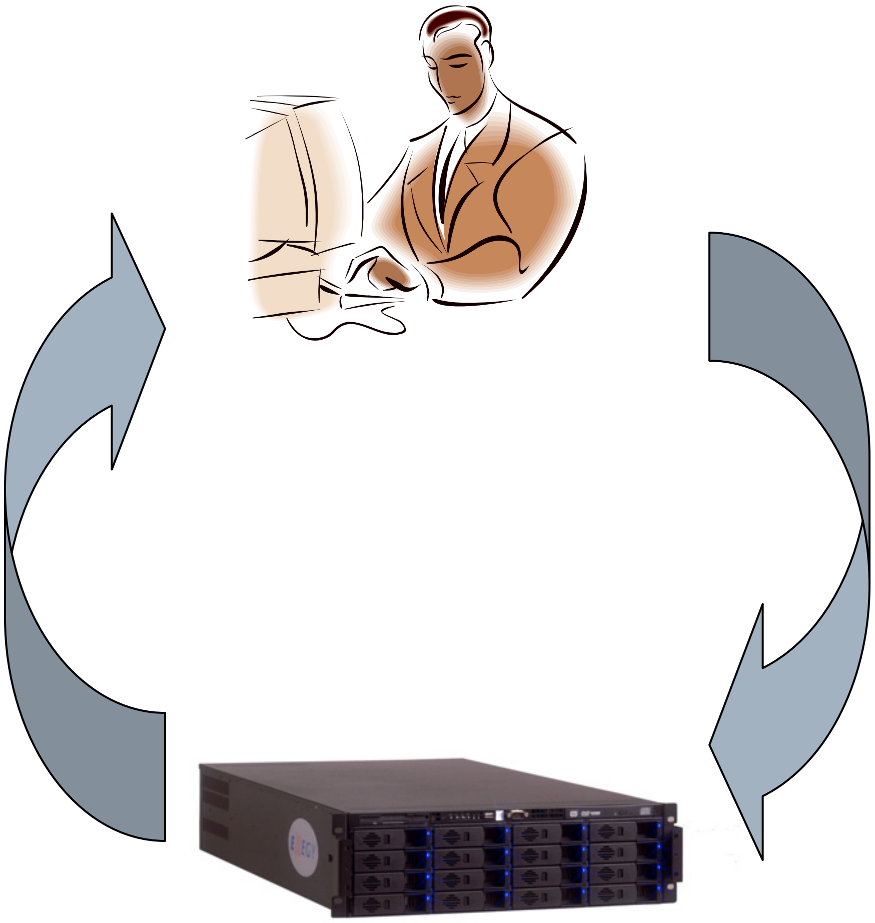
formulate
initial
query

Benefits of reconfiguration



formulate initial query

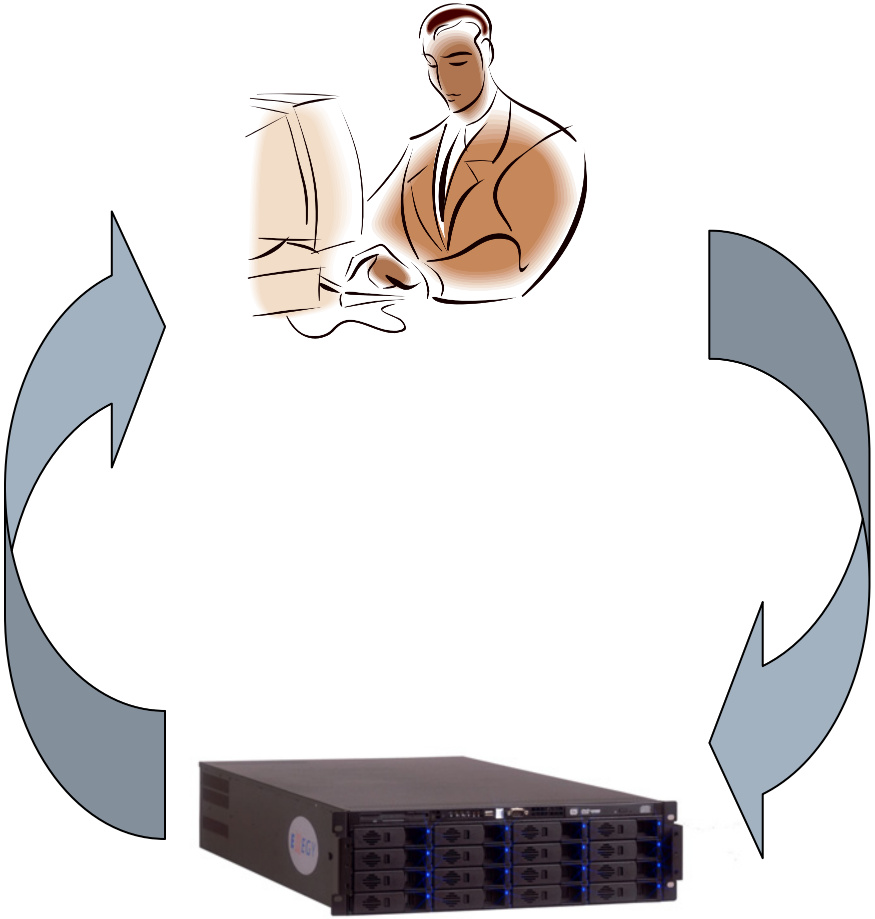
Benefits of reconfiguration



analyze
query
results

formulate
initial
query

Benefits of reconfiguration



analyze
query
results

formulate
revised
query

Query Options

Exact Search

- Literal keywords must match exactly
- Tens of thousands of keywords searched in one pass across the data

Approximate Search

- Wildcard characters
- Case insensitivity
- Character substitution up to specified bound

Regular Expression Search

- Full expressive power of finite-state machine recognizer

Exact Match Engine

Startup

- Hash keywords to a bit vector in FPGA
- Rabin-Karp hash functions

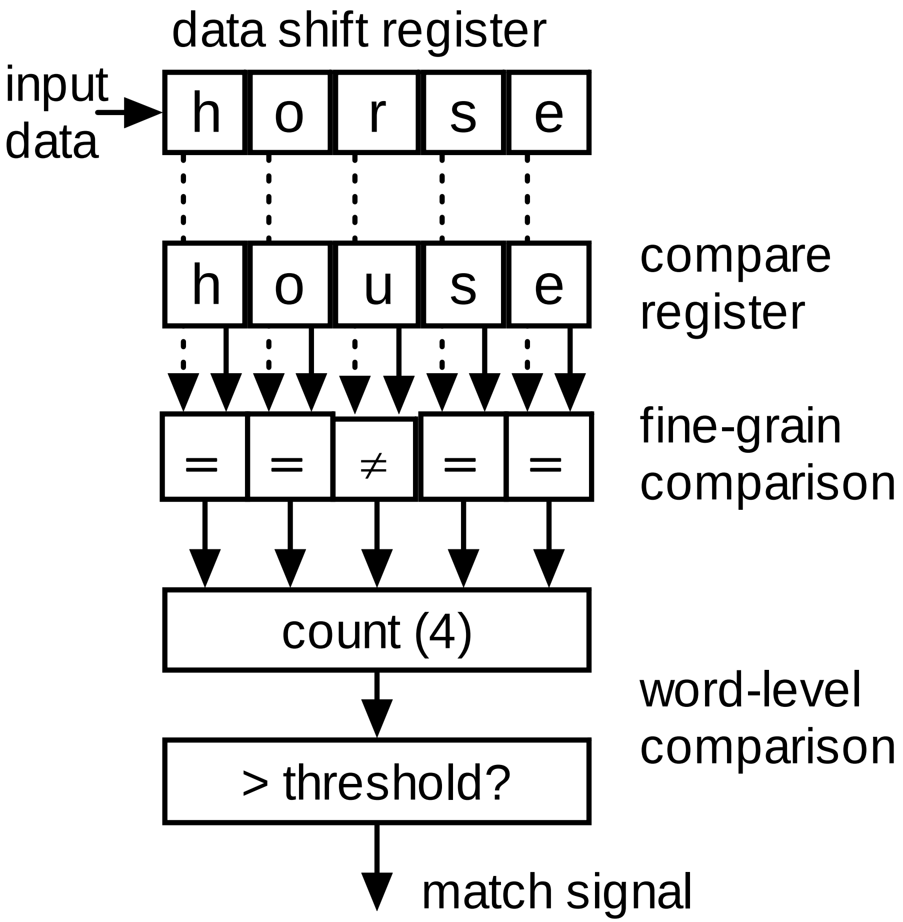
Run

- Stream text corpus from disk or network to FPGA
- Hash text to bit vector position
- Check position for keyword hit

Check

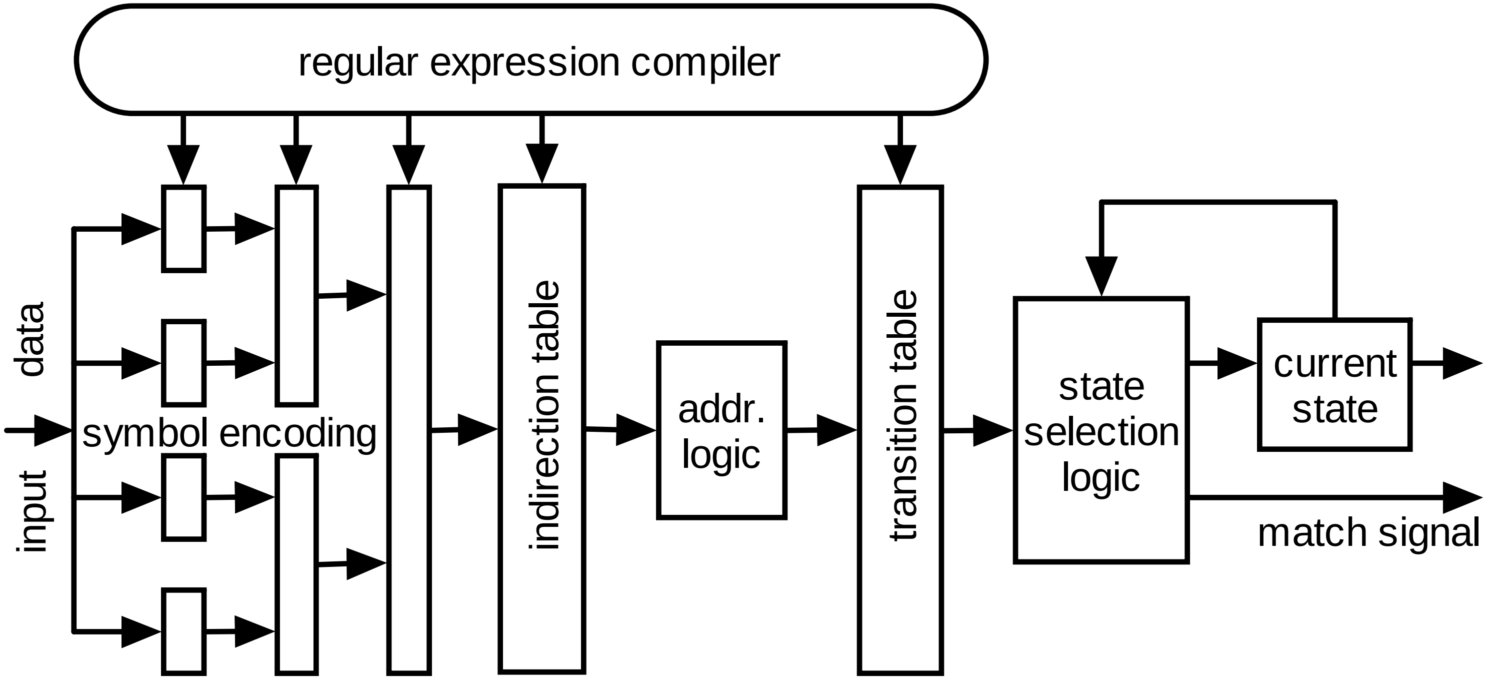
- False positives from hash collisions checked in software

Approximate Match Engine

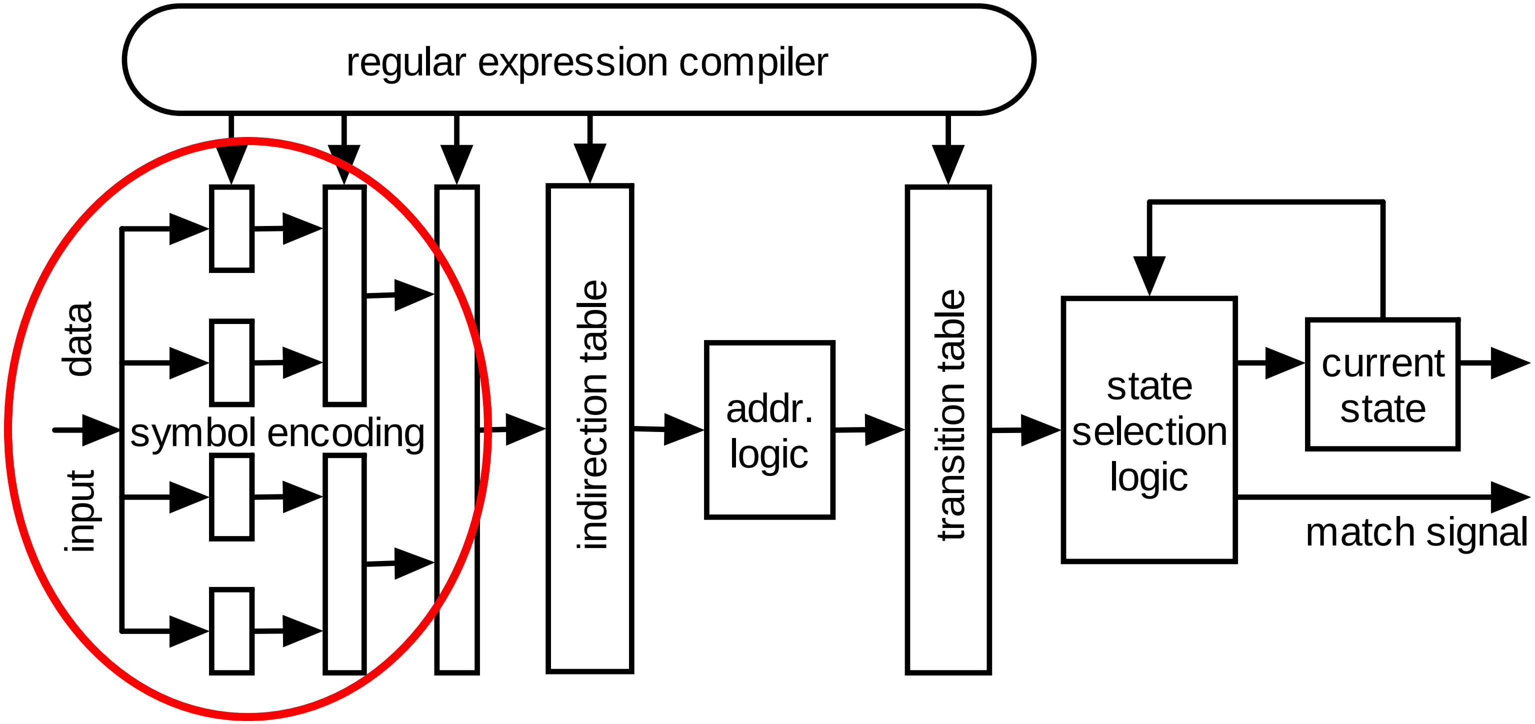


- Data shift register receives inbound text
- Compared with keyword at character level
- Count of matching characters is checked with threshold
- If character matches exceed threshold, keyword is a match

Regular Expression Engine

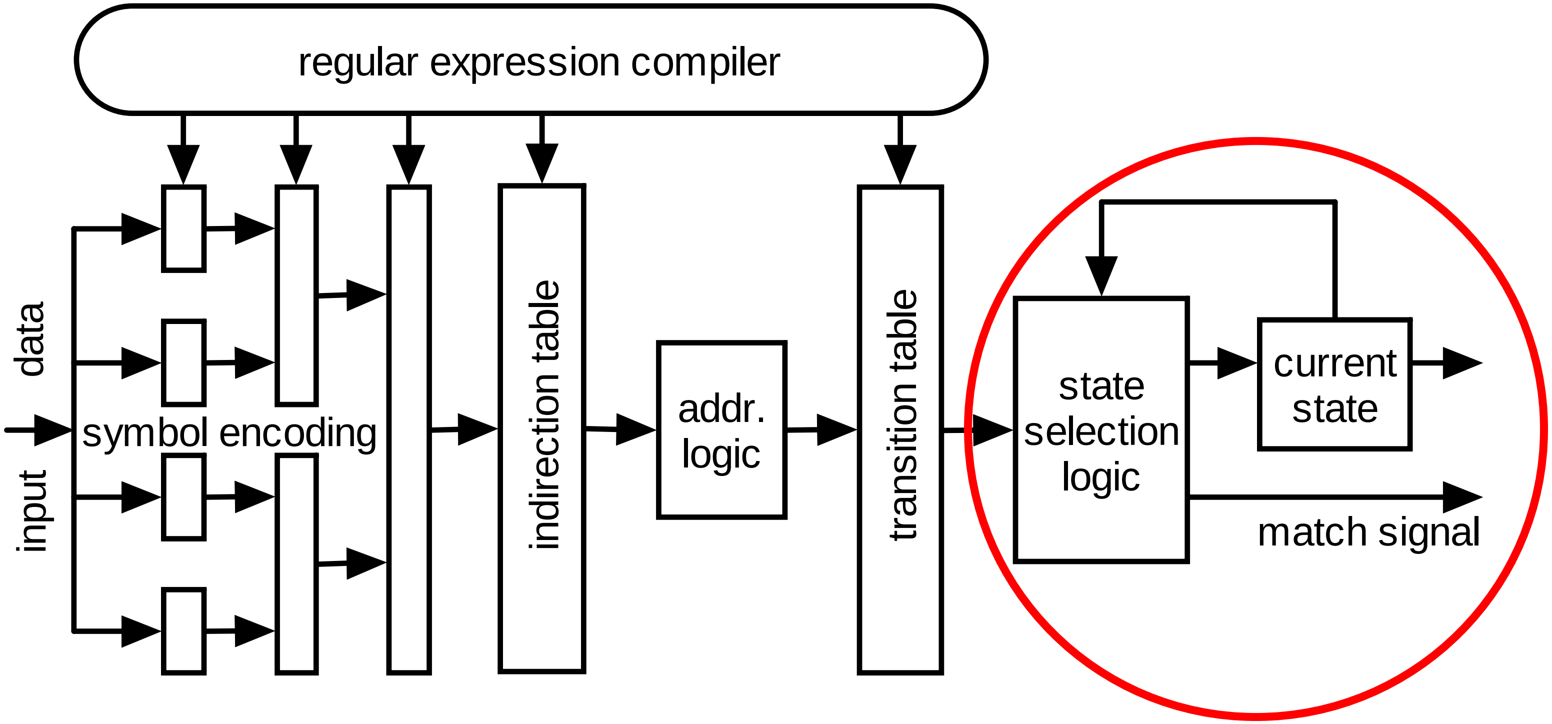


Regular Expression Engine



- Multi-character strings are combined into single symbol for finite state machine recognizer

Regular Expression Engine

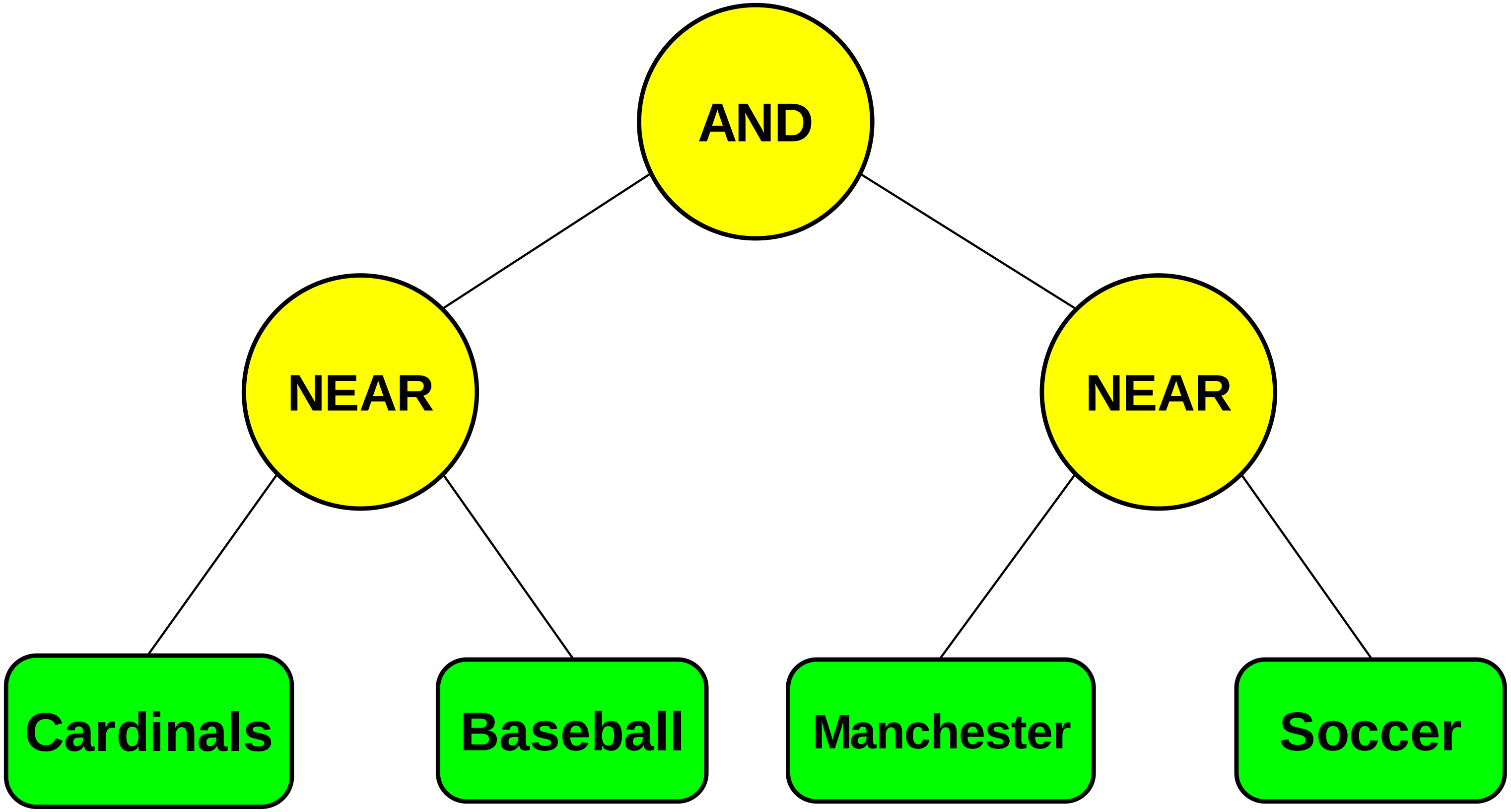


- Multi-character strings are combined into single symbol for finite state machine recognizer
- State dependent transitions are deferred to end of pipeline

EXEGY

Combining Operations

(Cardinals NEAR Baseball) AND (Manchester NEAR Soccer)



- Combining operations implemented in software
- Based on keyword hits from FPGA

Summary of 3 Hardware Search Engines

- Searching for individual terms, combining operations performed in software
- Three distinct engines supported:

  Exact match

  - Thousands of terms, 800 MB/s search rate

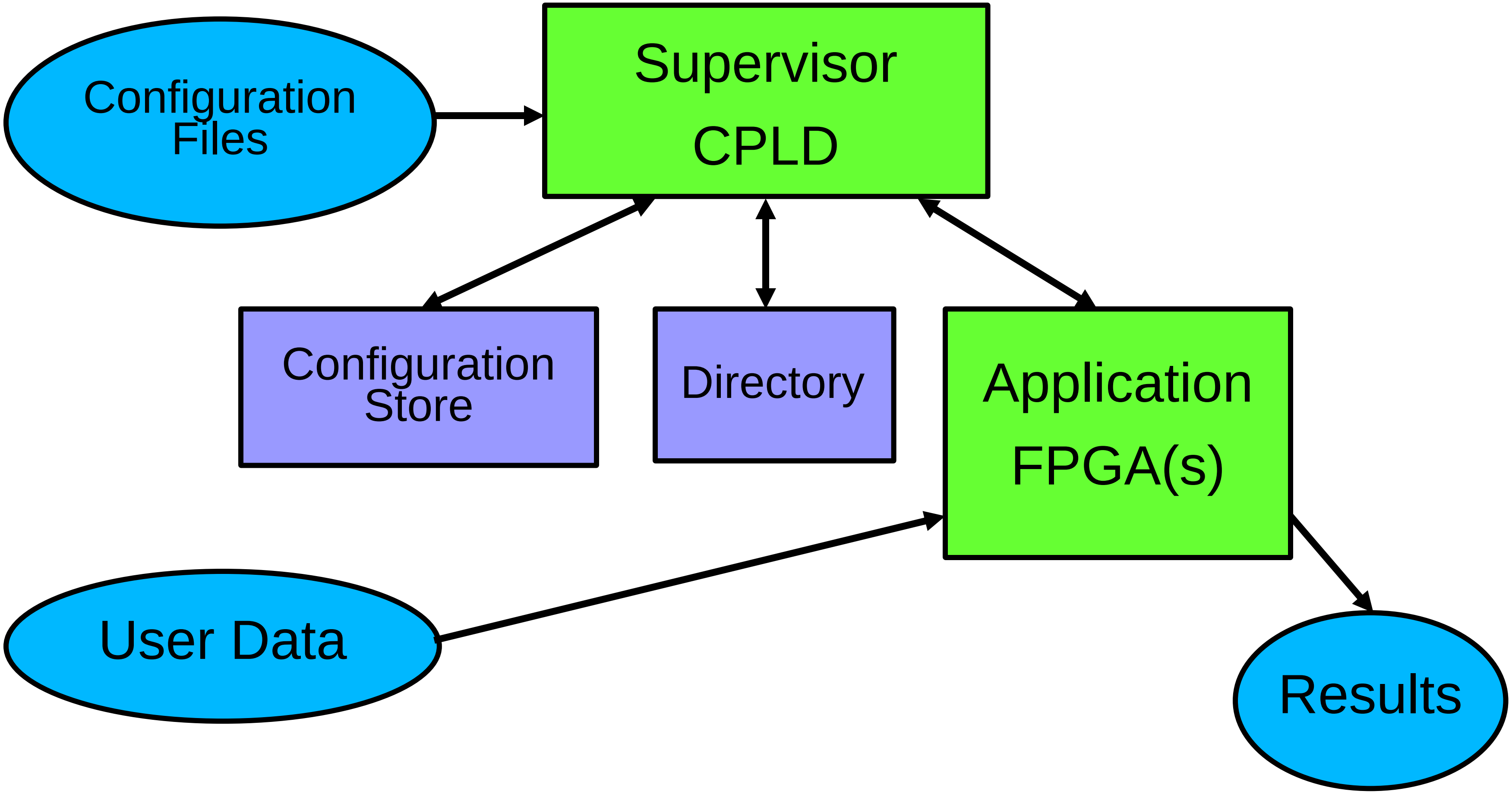
  Approximate match

  - Can trade off # of terms vs. characters per term, 800 MB/s search rate

  Regular expression search

  - Capable of ~50 expressions, 400 MB/s search rate
- Data source(s) can be local or remote

Managing FPGA Configuration



Supervisor Functions

Manage Configuration Store

- On-board non-volatile storage for configurations
- Supports multiple configuration files

Manage Directory

- Meta-data for configurations in configuration store

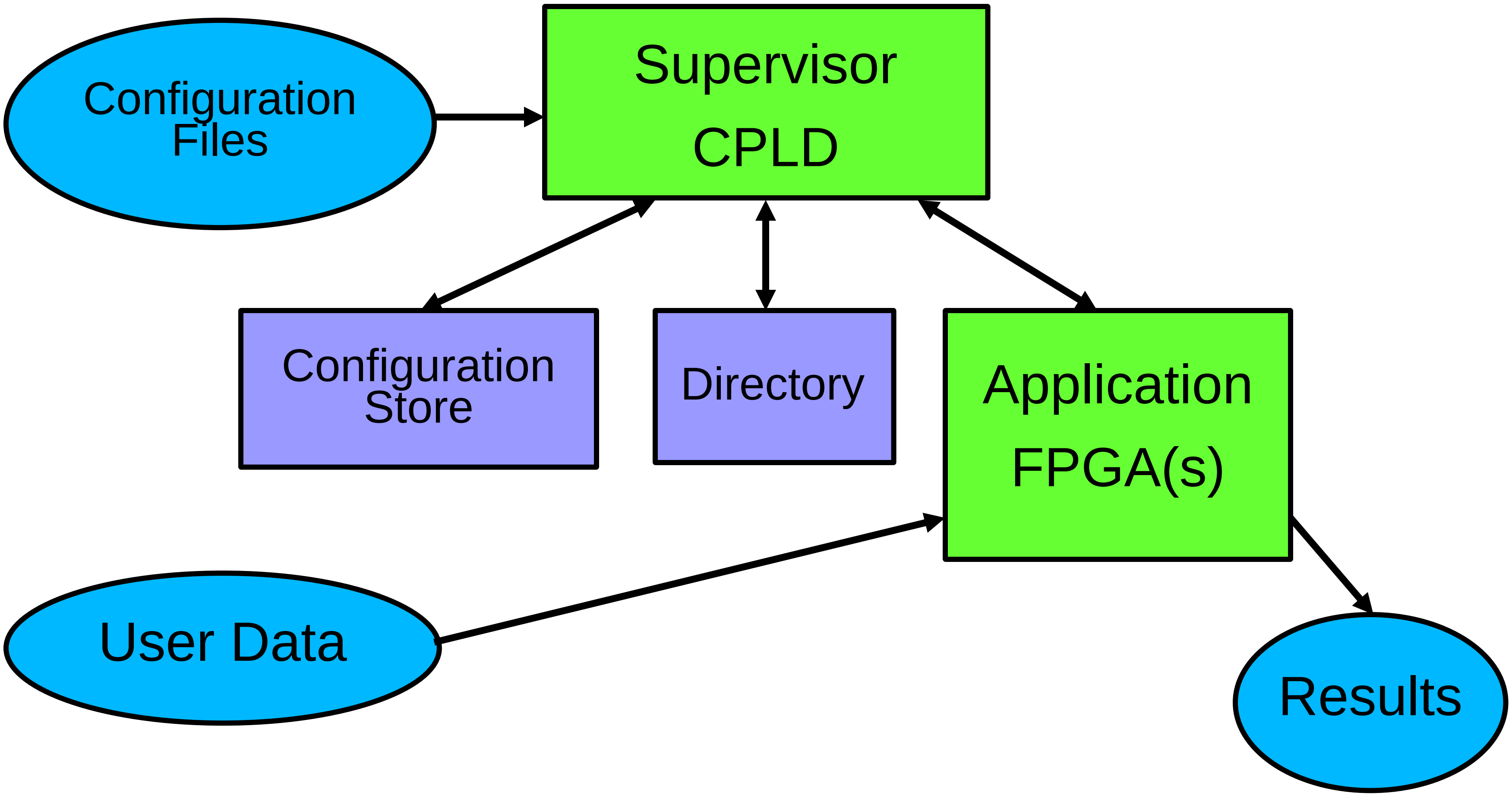
Load FPGA as instructed

- Reconfigure FPGA from specified config file currently in configuration store

Protection

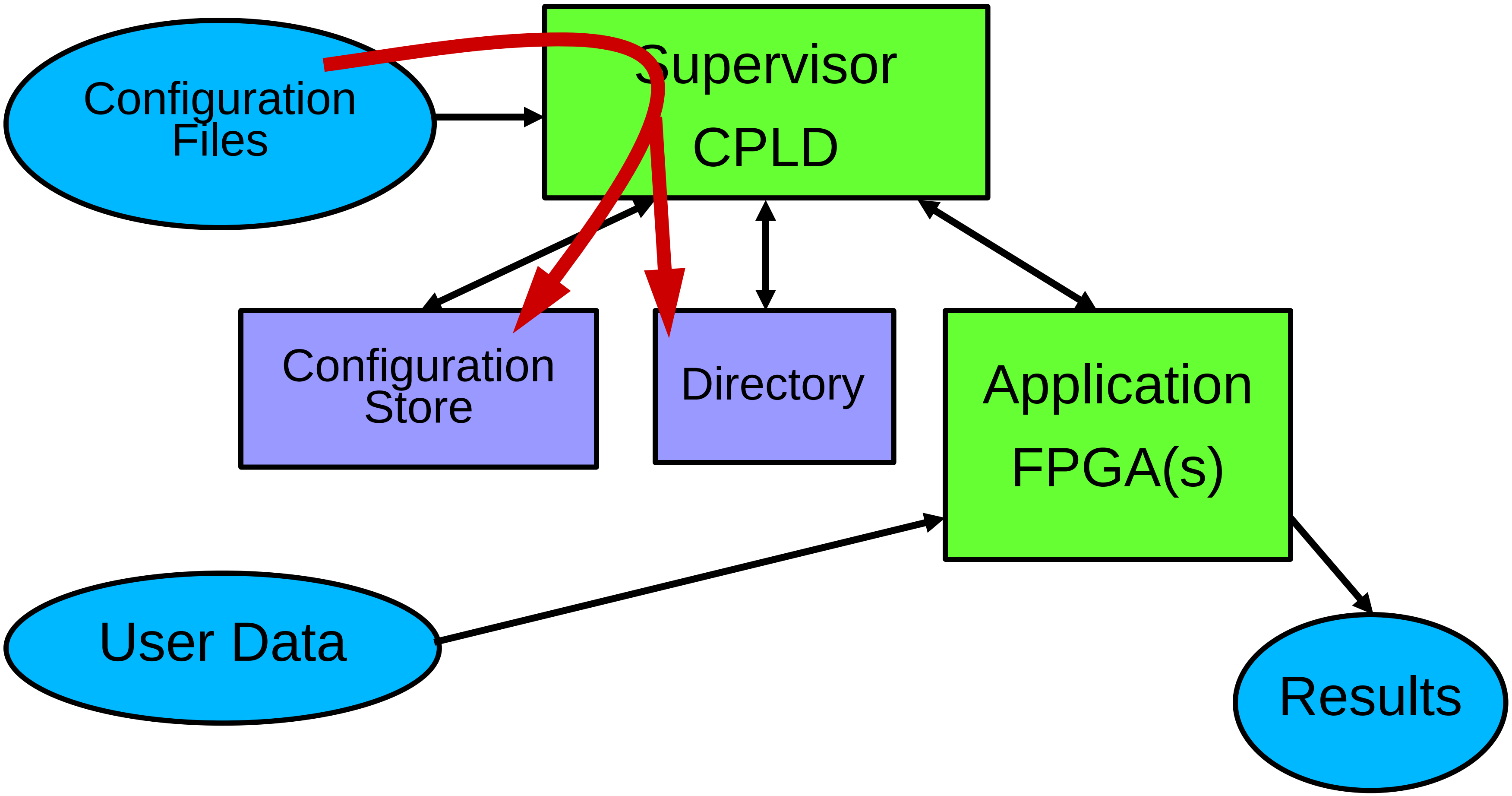
- Block data path during reconfiguration
- Check configuration is appropriate for that FPGA

Software Options: Insert



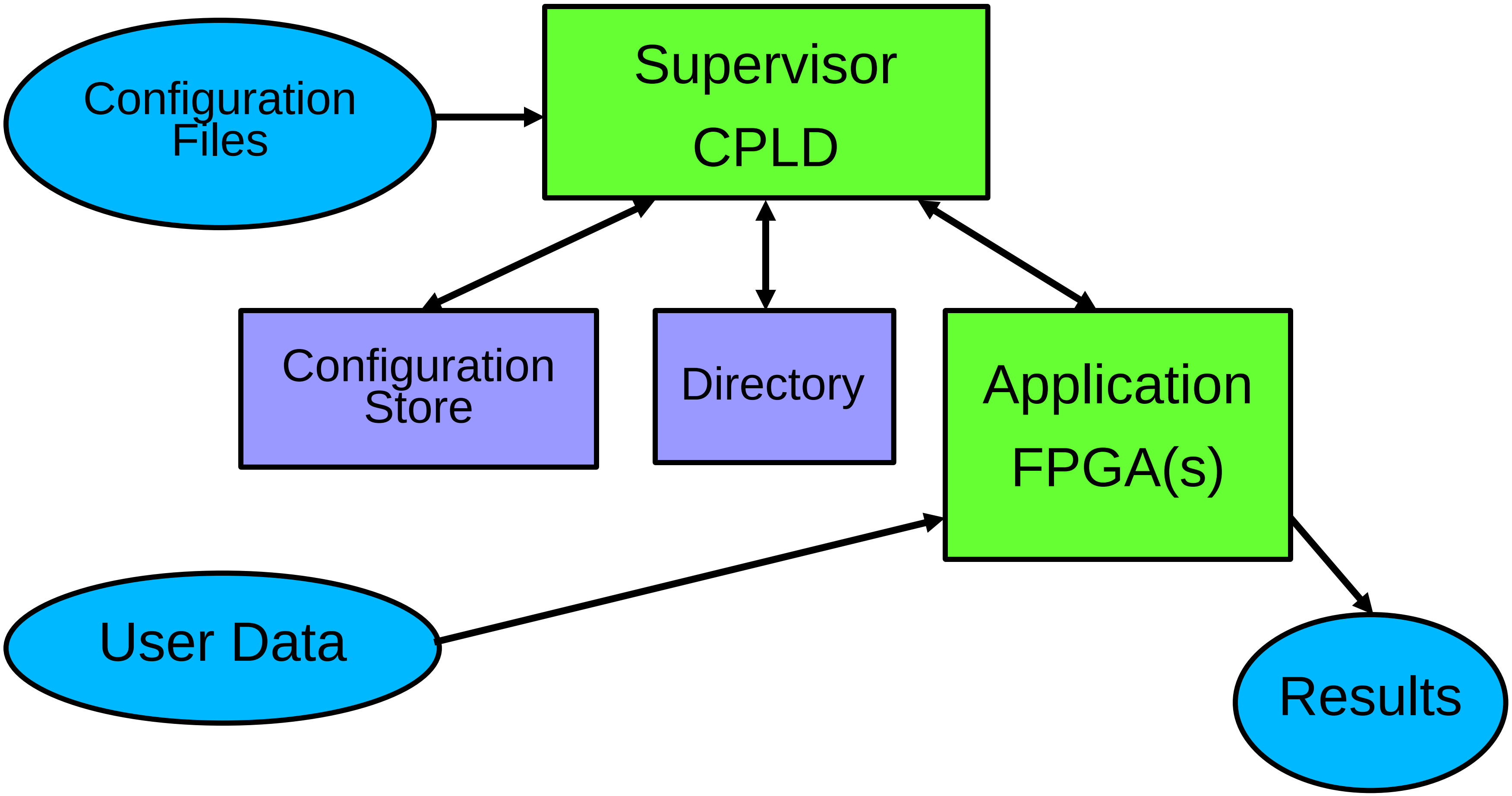
Place a configuration in the on-board store

Software Options: Insert



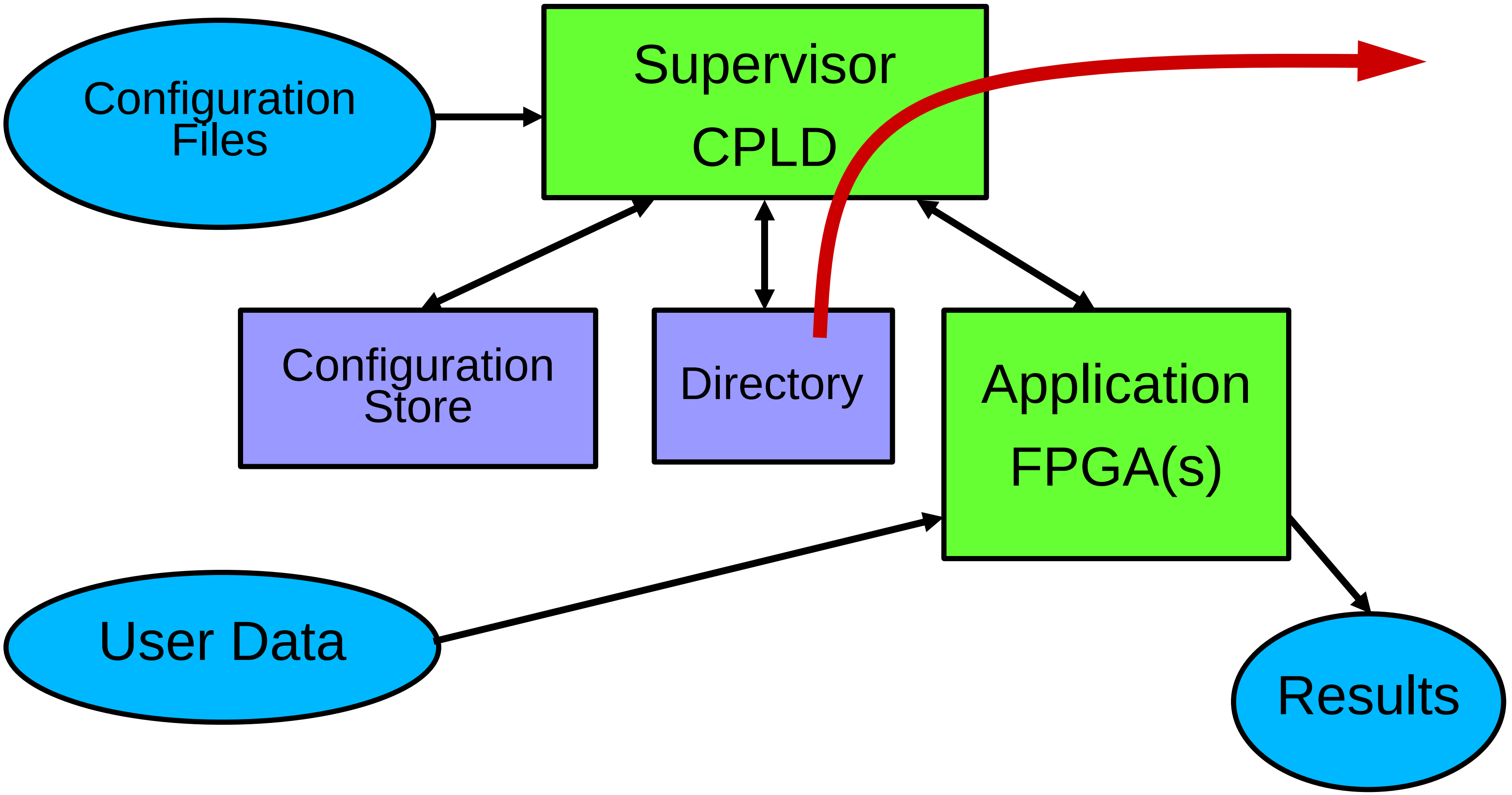
Place a configuration in the on-board store

Read Directory



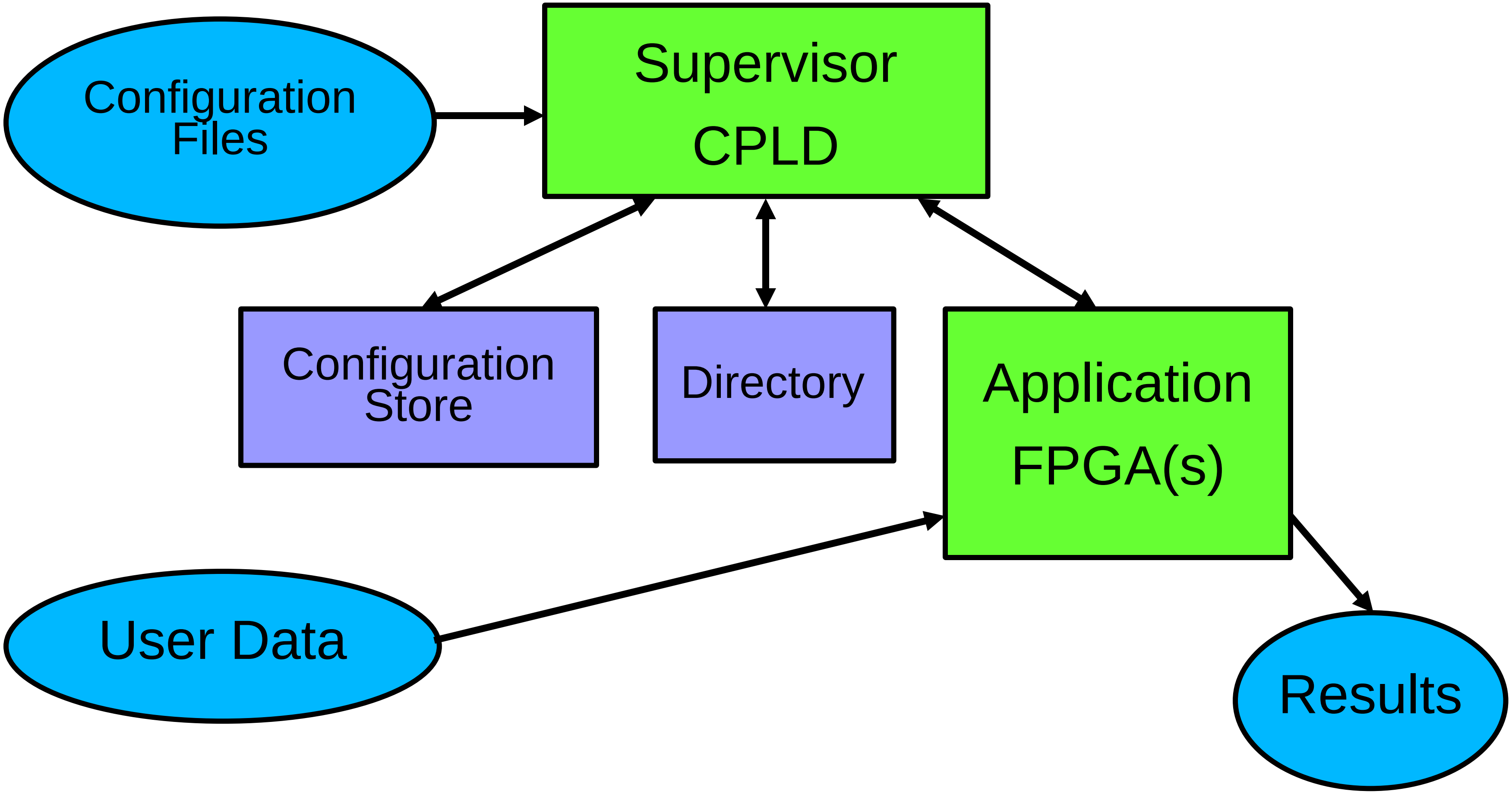
Query current directory contents

Read Directory



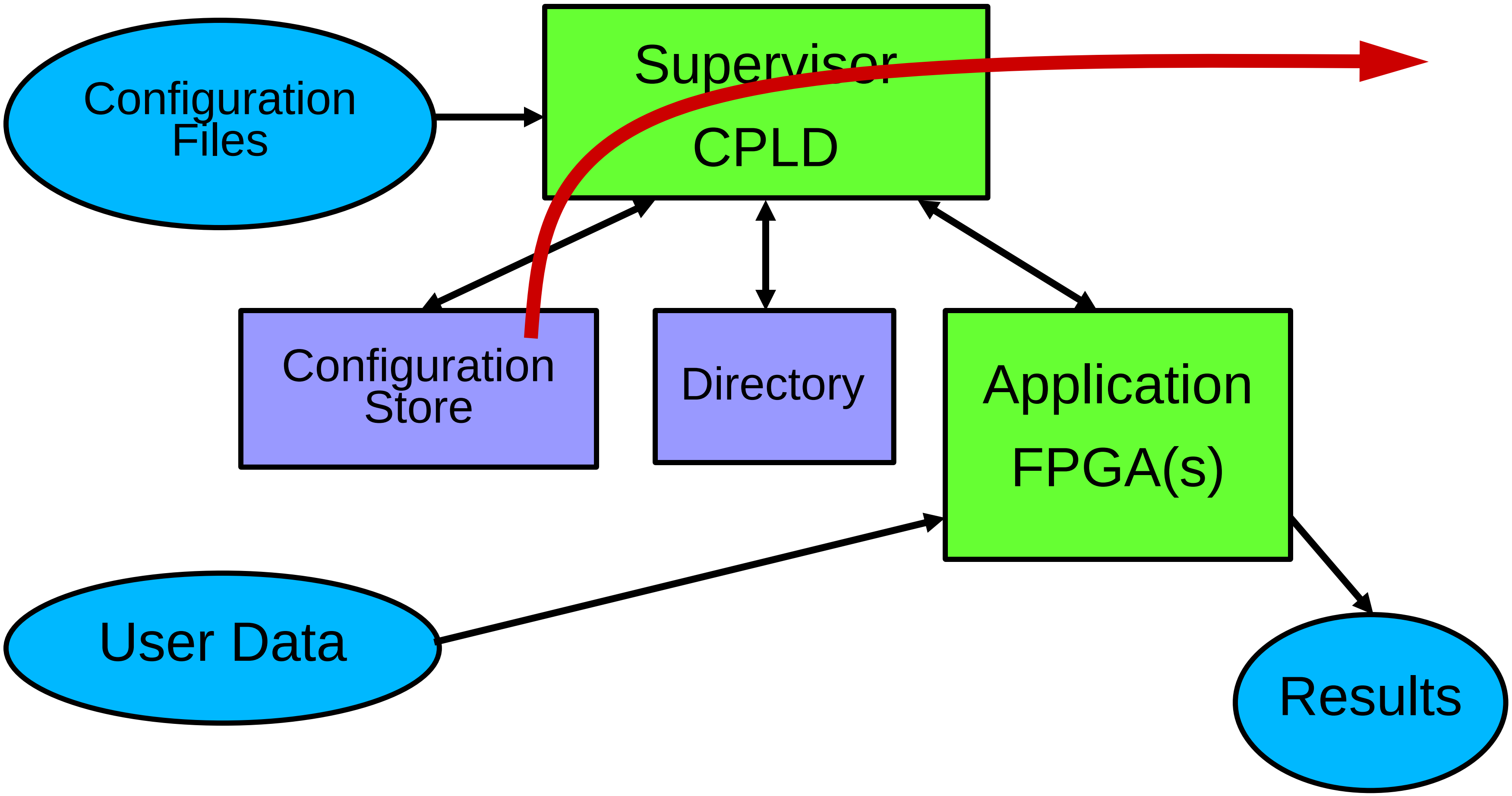
Query current directory contents

Read Configuration



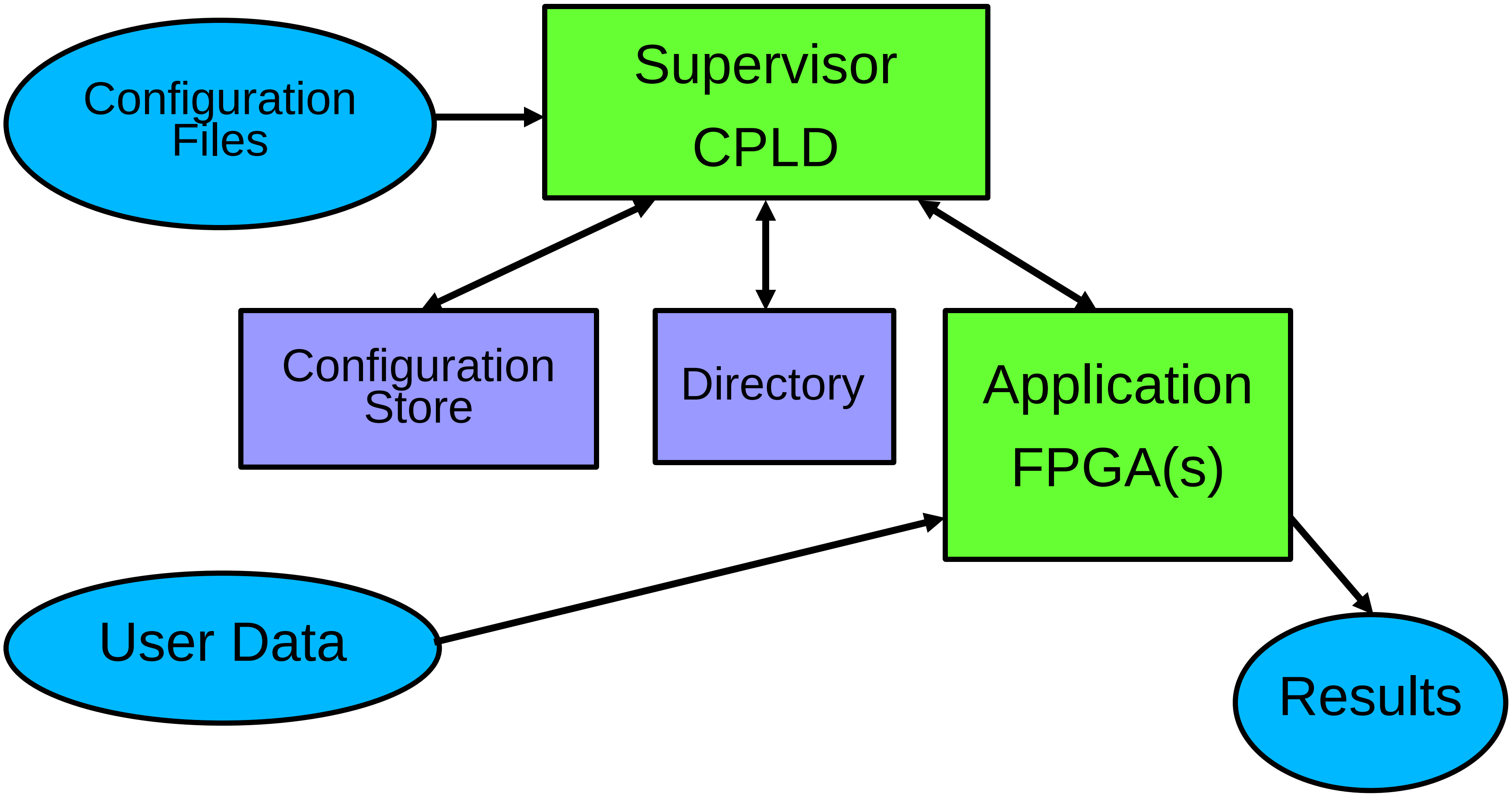
Primarily for verification and debugging purposes

Read Configuration



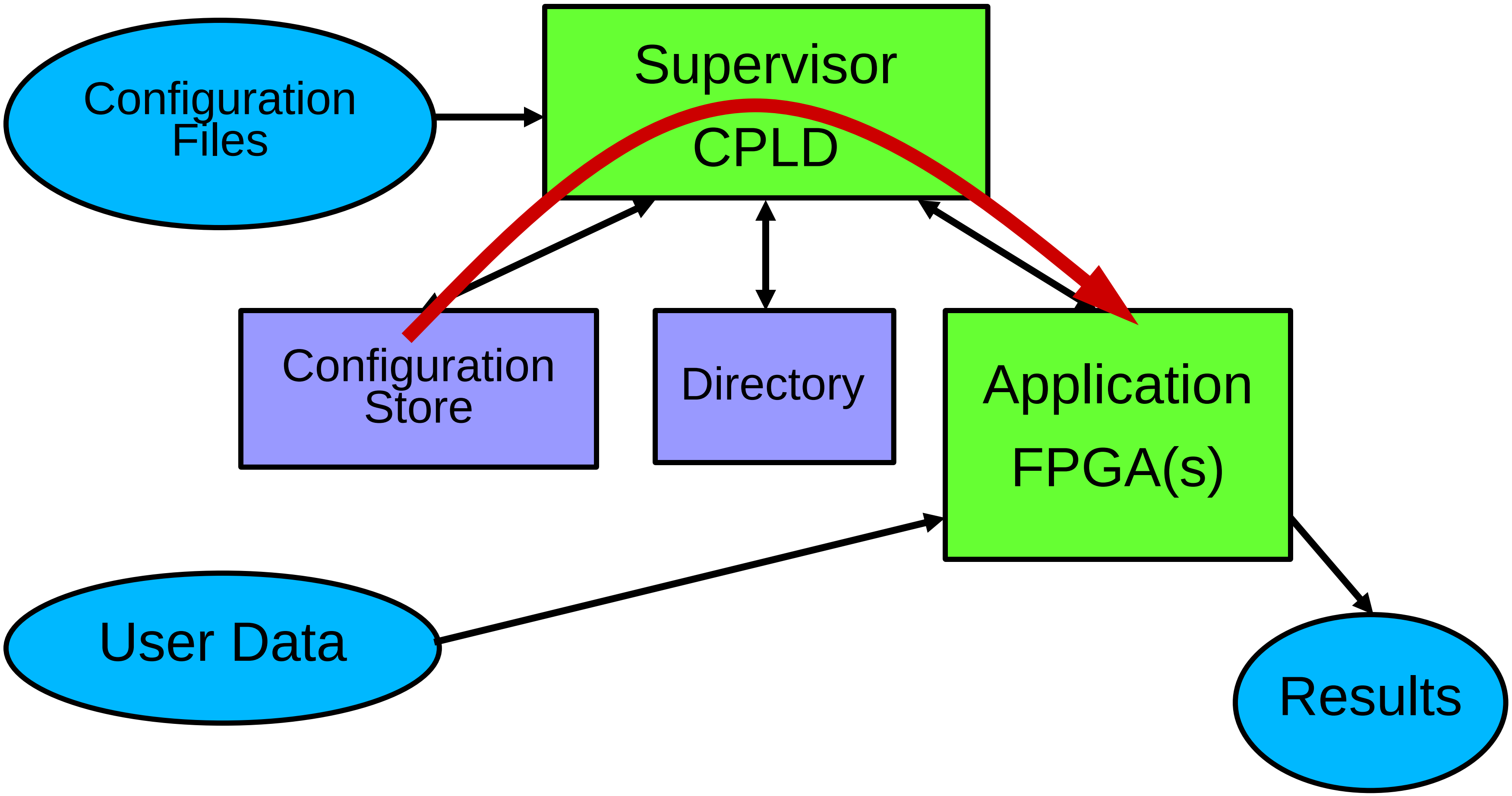
Primarily for verification and debugging purposes

Load



Reconfigure FPGA

Load



Reconfigure FPGA

Back to Text Search Application

Sequence of events:

1. User provides query and initiates search
2. Examining search terms, software selects appropriate engine
3. Load configuration in FPGA, concurrently queue up data from source
4. Load search terms into engine
5. Stream data through engine
6. Process hits that return, performing combining operations
7. Return results to user

Comments

Benefits

- Application software chooses appropriate FPGA engine
- Engine is tailored to problem at hand

Concerns

- Heterogeneous query
  - Requires multiple engines or multiple data passes
- Configuration overhead
  - 20 ms is longer than we would like
  - However, it’s not out of line with startup times required for disk access

Summary

- Exegy A2000 appliance supports dynamic reconfiguration of application FPGAs
- Exegy TextMiner application exploits dynamic reconfiguration for text search
- 3 distinct search engines: exact, approximate, and regular expression
- FPGA configuration is concurrent with initial data reads to mask latency
- Result is a true exploitation of the physical ability to reconfigure FPGAs on the fly