CRICOS PROVIDER 00123M

Faculty of SET / School of Computer Science

COMP SCI 1400 AI Technologies

AI Projects

Dr. Kamal Mammadov

Outline

- ELIZA
- Siri
- MYCIN
- Cyc
- Deep Blue
- Alpha Go
- ImageNet

ElIZA

- By Joseph Weizenbaum, 1966

```
> Hello, I am Eliza.
* I am afraid.
> How long have you been afraid?
* Since midterms.
> Tell me more...
* I am afraid that I will fail
my course.
> Do you enjoy being afraid that
you will fail your course?
```

ELIZA

- Two parts:
  - ✓ natural language processing
  - ✓ response generation

- Eliza simulated conversation by using **pattern matching** and **substitution.**

ELIZA

- Use Regular Expression

Examples:
•.at matches any three-character string ending with "at", including "hat", "cat", and "bat".
•[hc]at matches "hat" and "cat".
•[^b]at matches all strings matched by .at except "bat".
•[^hc]at matches all strings matched by .at other than "hat" and "cat".
•^[hc]at matches "hat" and "cat", but only at the beginning of the string or line.
•[hc]at$ matches "hat" and "cat", but only at the end of the string or line.
•\[.\] matches any single character surrounded by "[" and "]" since the brackets are escaped, for example: "[a]" and "[b]".
•s.* matches s followed by zero or more characters, for example: "s" and "saw" and "seed".

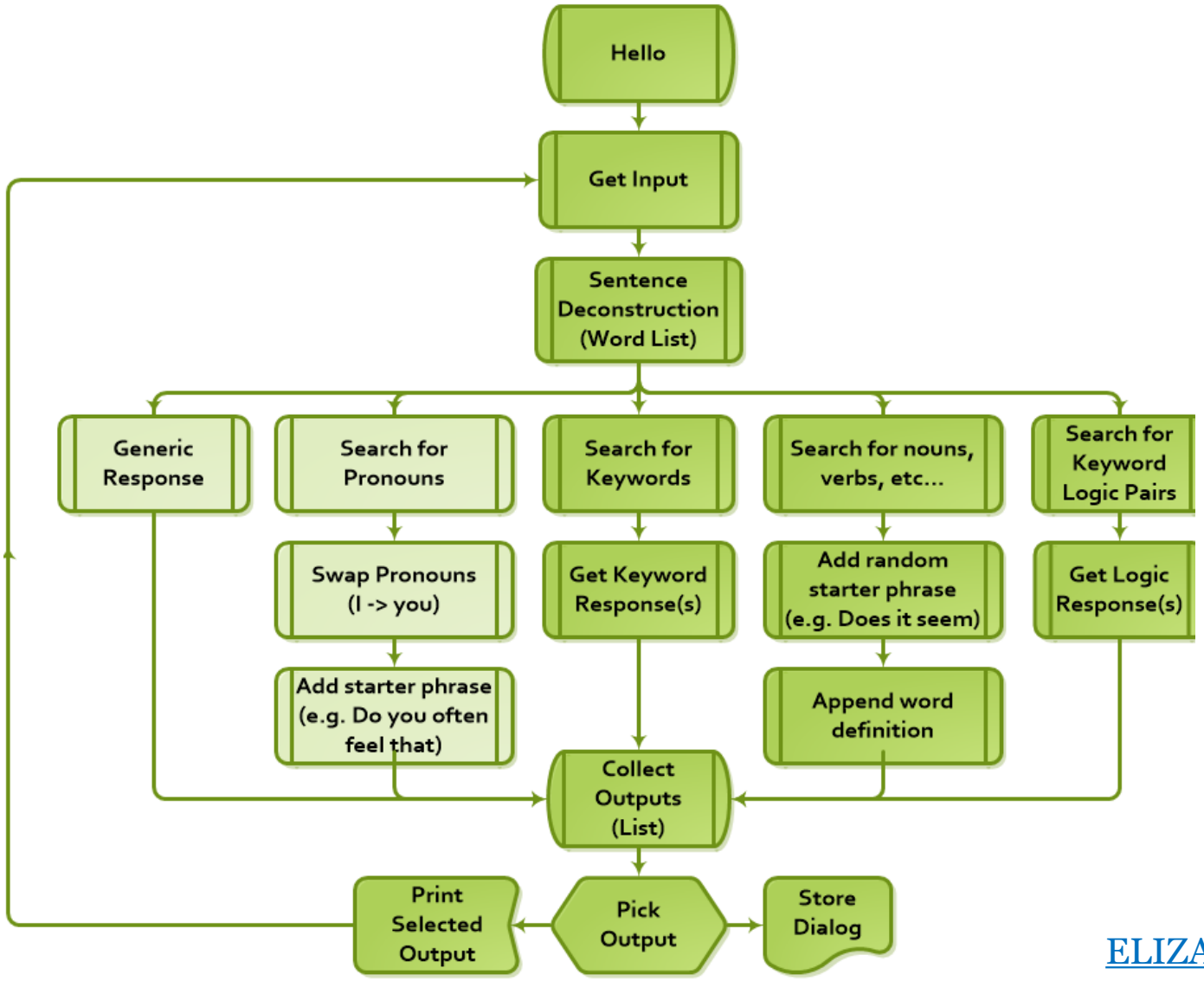
- Pre-set Q-A pair templates.

```
pairs = (
    (
        r"I need (.*)",
        (
            "Why do you need %1?",
            "Would it really help you to get %1?",
            "Are you sure you need %1?",
        ),
    ),
    (
        r"Why don\'t you (.*)",
        (
            "Do you really think I don't %1?",
            "Perhaps eventually I will %1.",
            "Do you really want me to %1?",
        ),
    ),
    (
        r"Why can\'t I (.*)",
        (
            "Do you think you should be able to %1?",
            "If you could %1, what would you do?",
            "I don't know -- why can't you %1?",
            "Have you really tried?",
        ),
    ),
    (
        r"I can\'t (.*)",
        (
            "How do you know you can't %1?",
            "Perhaps you could %1 if you tried.",
            "What would it take for you to %1?",
        ),
    ),
    (
        r"I am (.*)",
        (
            "Did you come to me because you are %1?",
            "How long have you been %1?",
            "How do you feel about being %1?",
        ),
    ),
    (
        r"I\'m (.*)",
        (
            "How does being %1 make you feel?",
```

```
reflections = {
    "i am": "you are",
    "i was": "you were",
    "i": "you",
    "i'm": "you are",
    "i'd": "you would",
    "i've": "you have",
    "i'll": "you will",
    "my": "your",
    "you are": "I am",
    "you were": "I was",
    "you've": "I have",
    "you'll": "I will",
    "your": "my",
    "yours": "mine",
    "you": "me",
    "me": "you",
}
```

NLTK

ELIZA



ELIZA BICKFORD

SIRI

- Siri is an intelligent personal assistant and knowledge navigator with a voice-interface in Apple's IOS and MacOS
- It was created by Apple in 2010 and is a breakthrough in AI application.
  - Speech recognition + NLP

Siri

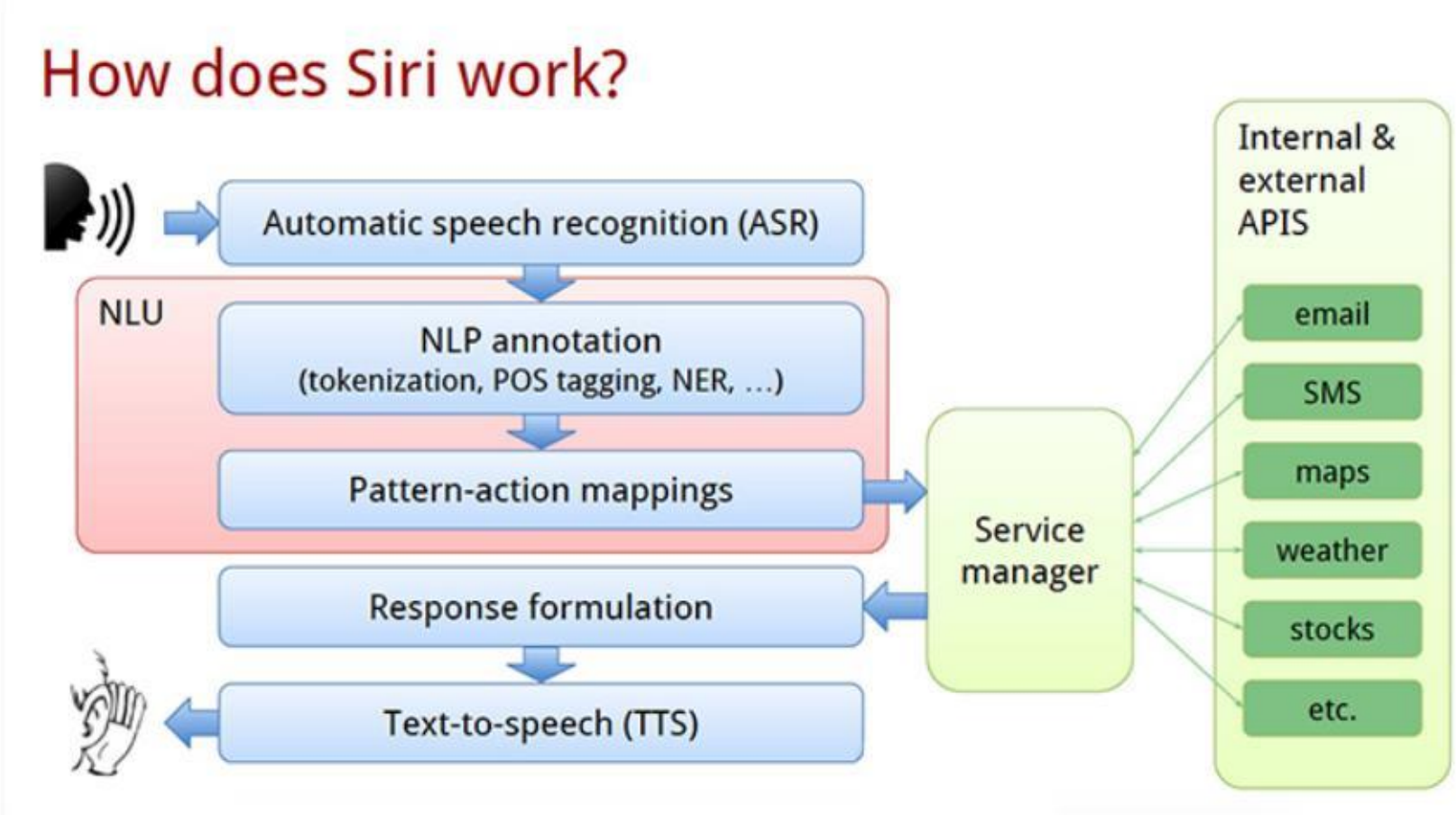
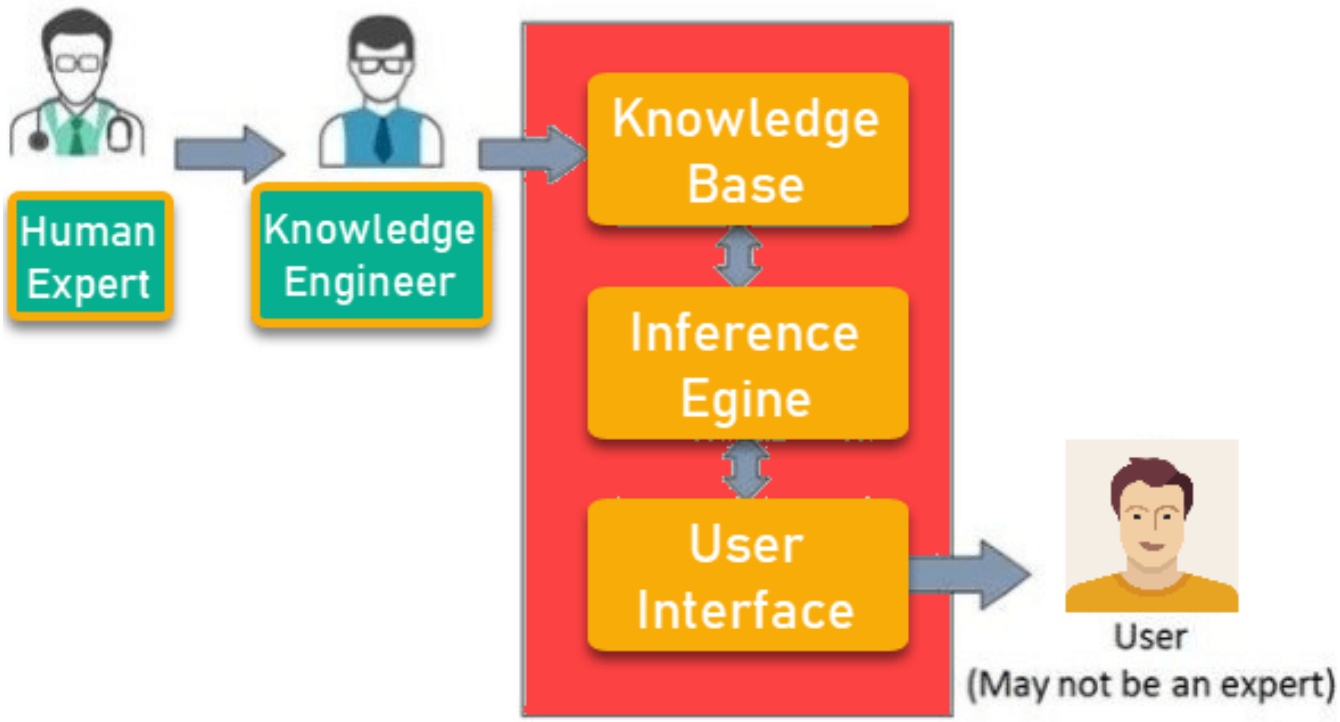


Image source

Expert Systems

- In AI, an **expert system** is a computer system that emulates the decision-making ability of a human expert.
- The first expert systems were created in the 1970's.



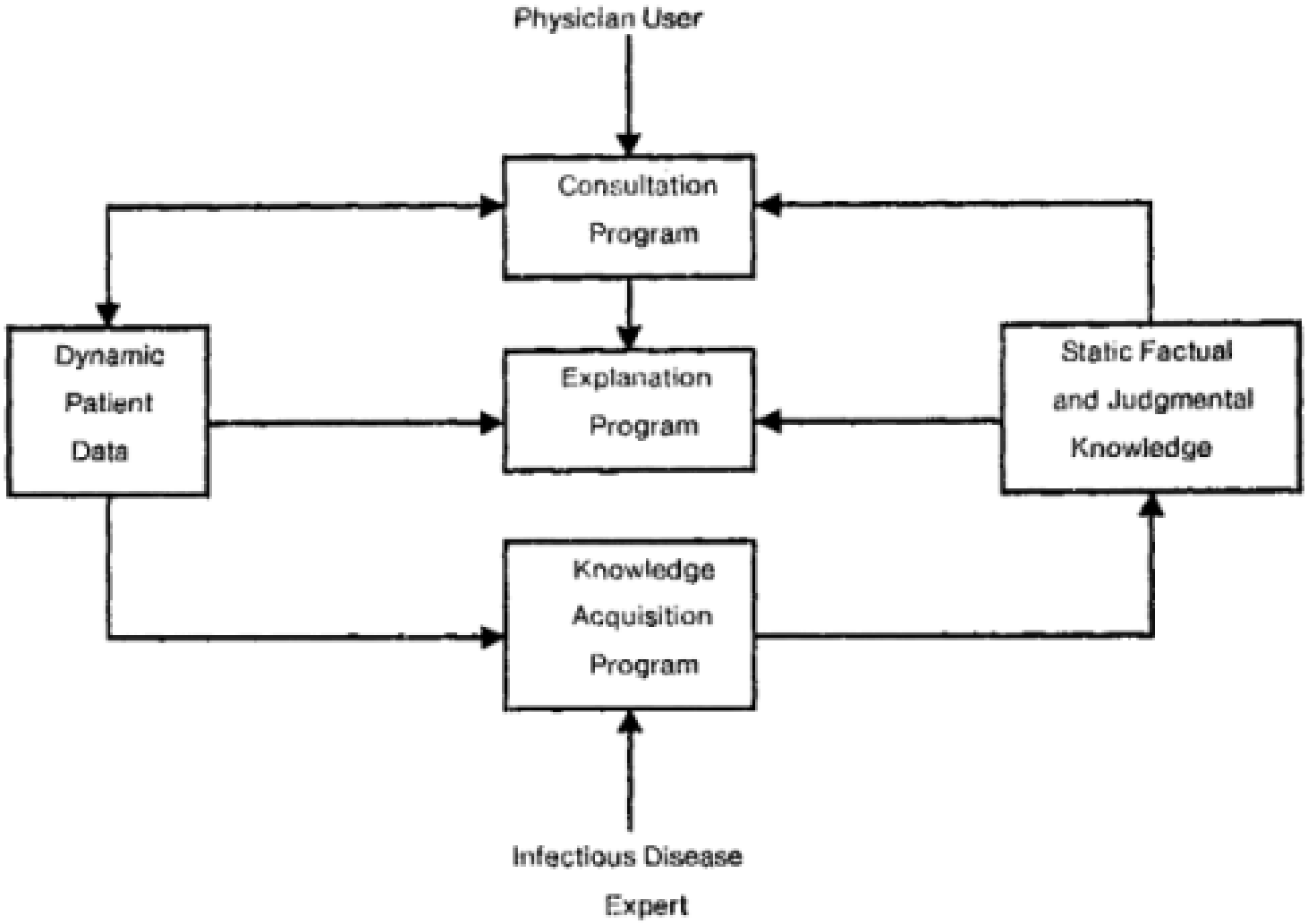
MYCIN

- One of the earliest expert systems, 1970s by Stanford University.
- For diagnosis and treatment recommendation for various acute blood infections.
- MYCIN was never actually used in practice due to ethical issues and integration limitations, but it outperformed the infectious disease experts.

MYCIN

- Use a fairly simple inference engine and a knowledge base of 600 rules.
- It worked by querying the user through a long series of simple yes/no or textual questions.
- Output: a list of possible culprit bacteria, its confidence in each diagnosis, the reasoning (referring to individual questions and answers) being each diagnosis, and its recommended course of drug treatment.

MYCIN Architecture



MYCIN

- Consultation examples

```
--------PATIENT-1--------
1)  Patient's name: (first-last)
**  FRED SMITH
2)  Sex:
**  MALE
3)  Age:
** 55
4)  Have you been able to obtain positive cultures from a site
    at which Fred Smith has an infection?
**  YES
  --------INFECTION-1--------
  5)  What is the infection?
  **  PRIMARY-BACTEREMIA
  6)  Please give the date and approximate time when signs or symptoms
      of the primary-bacteremia (INFECTION-1) first appeared. (mo/da/yr)
  **  MAY 5, 1975
  The most recent positive culture associated with the
  primary-bacteremia (INFECTION-1) will be referred to as:
    --------CULTURE-1--------
    7)  From what site was the specimen for CULTURE-1 taken?
    ** BLODD
    = BLOOD
    8)  Please give the date and time when this blood culture
        (CULTURE-1) was obtained. (mo/da/yr)
    ** MAY 9, 1975
    The first significant organism from this blood culture
    (CULTURE-1) will be referred to as:
      --------ORGANISM-1--------
      9)  Enter the identity of ORGANISM-1:
      **  UNKNOWN
            {typically the identity is not known and must be inferred}
      10)  Is ORGANISM-1 a rod or coccus (etc.):
      **  ROD
      11)  The gram stain of ORGANISM-1:
      **  GRAMNEG
```

MYCIN

Rule example:

If 1) the gram stain of the organism is gram negative,
and 2) the morphology of the organism is rod
and 3)the aerobicity of the organism is anaerobic,

Then there is suggestive evidence (.7) that the identity of the organism is Bacteroides.

Cyc

- “Cyc has the world’s broadest and deepest common sense knowledge base (KB), by orders of magnitude. The KB is not a database – it consists of real world axioms that Cyc uses to reason about the world and understand your data. Cyc’s KB includes more than 10,000 predicates, millions of collections and concepts, and more than 25 million assertions. ” - from official website

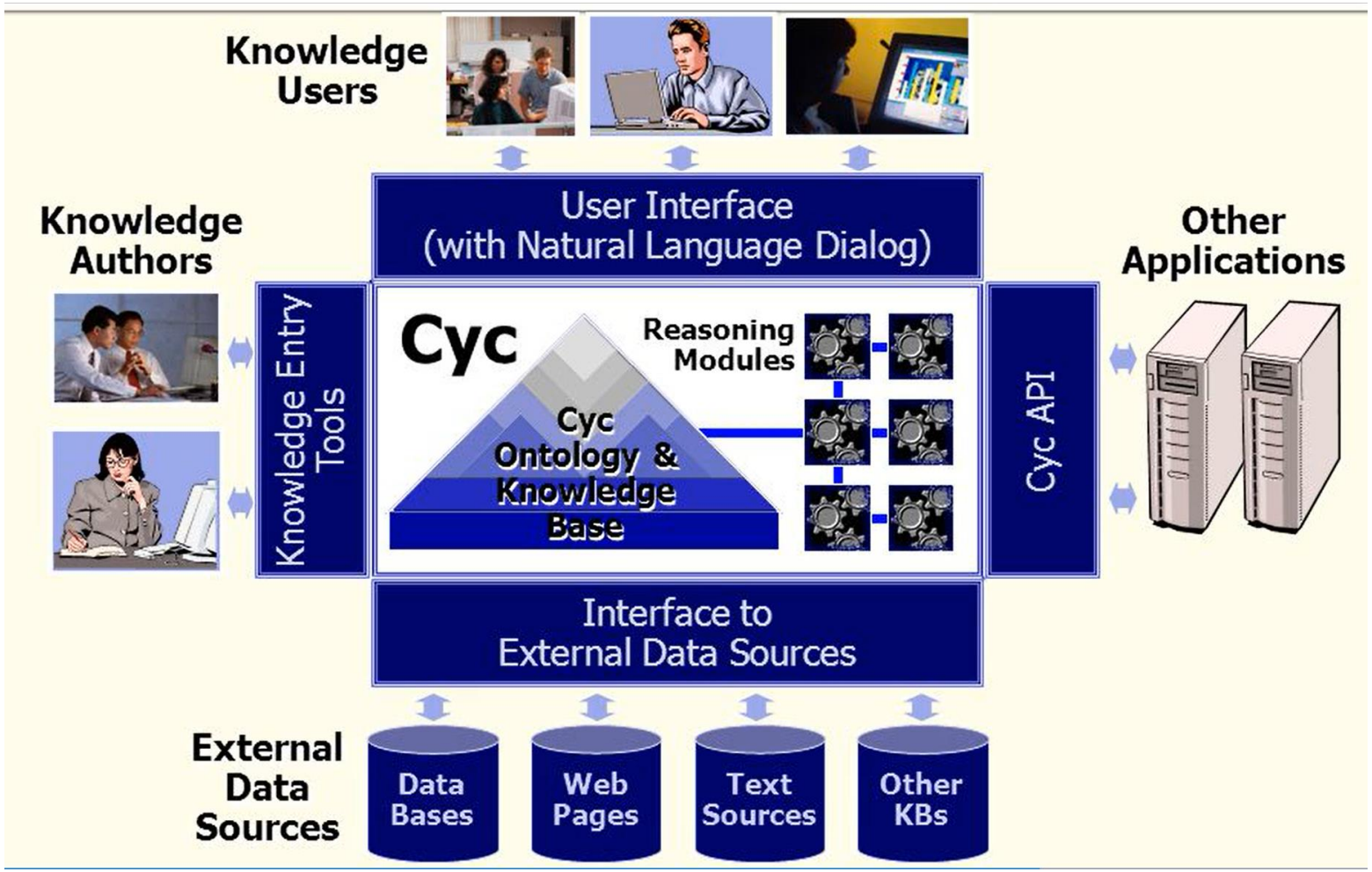
Cyc

- Founded in 1994, by Stanford professor Doug Lenat.
- Components:
  - Knowledge base
  - Inference engine
  - Representation language
  - Natural language subsystems
  - Semantic integration
  - Developer toolsets

Cyc

- The Cyc knowledge base consists of a collection of terms and assertions written in Cyc's logical language Cycl.
  - Assertions include both simple ground assertions and rules which relate the terms in the collection.
- The Cyc inference engine performs general logical deductions.
- The Cyc representation language:
  - Constants. e.g., #$MapleTree,
  - Variables. e.g., ?X
  - Predicates. E.g., #$isa..
- Cyc-NL subsystems:
  - Lexicon, Syntactic parser, Semantic Interpreter

Cyc Reasoning System



CSCI 8986 UNO Aruna Weerakoon

OpenCyc

- OpenCyc is a subset of Cyc, 2002-2017.
- A version of OpenCyc, 4.0, was released in June 2012. OpenCyc 4.0 included The OpenCyc 4.0 knowledge base contained 239,000 concepts and 2,093,000 facts.
- OpenCyc was the first large-scale, open-source knowledge base provided in OWL format. OpenCyc preceded Wikipedia in a form usable by the semantic Web.

IBM Deep Blue

- Deep Blue was IBM's chess-playing computer. It was the first computer system to win against the world chess champion (1996).

IMAGE: YVONNE HEMSEY/GETTY IMAGES

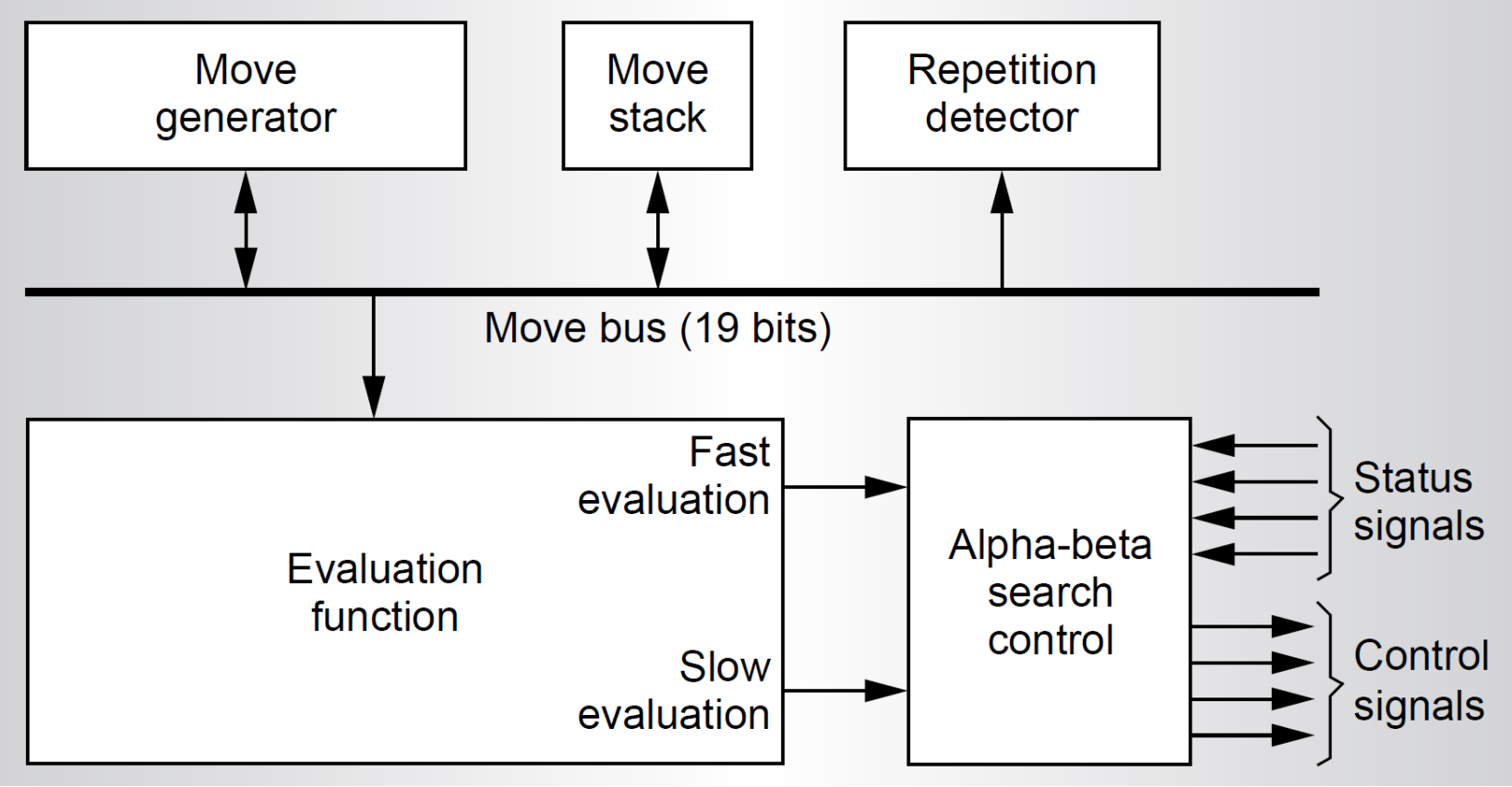
VS

IMAGE: STAN HONDA/AFP/GETTY IMAGES

IBM Deep Blue

- "Chip Test" at Carneige Mellon University by Feng-Hsuing Hsu -> "Deep Thought"-> "Deep Thought" -> "Deep Blue"
- Components:
  - Hardware: 30 IBM RS/6000 processors, each has 16 chess chips – 480 chess chips in total.
  - Algorithm: Evaluation functions, alpha-beta search algorithm

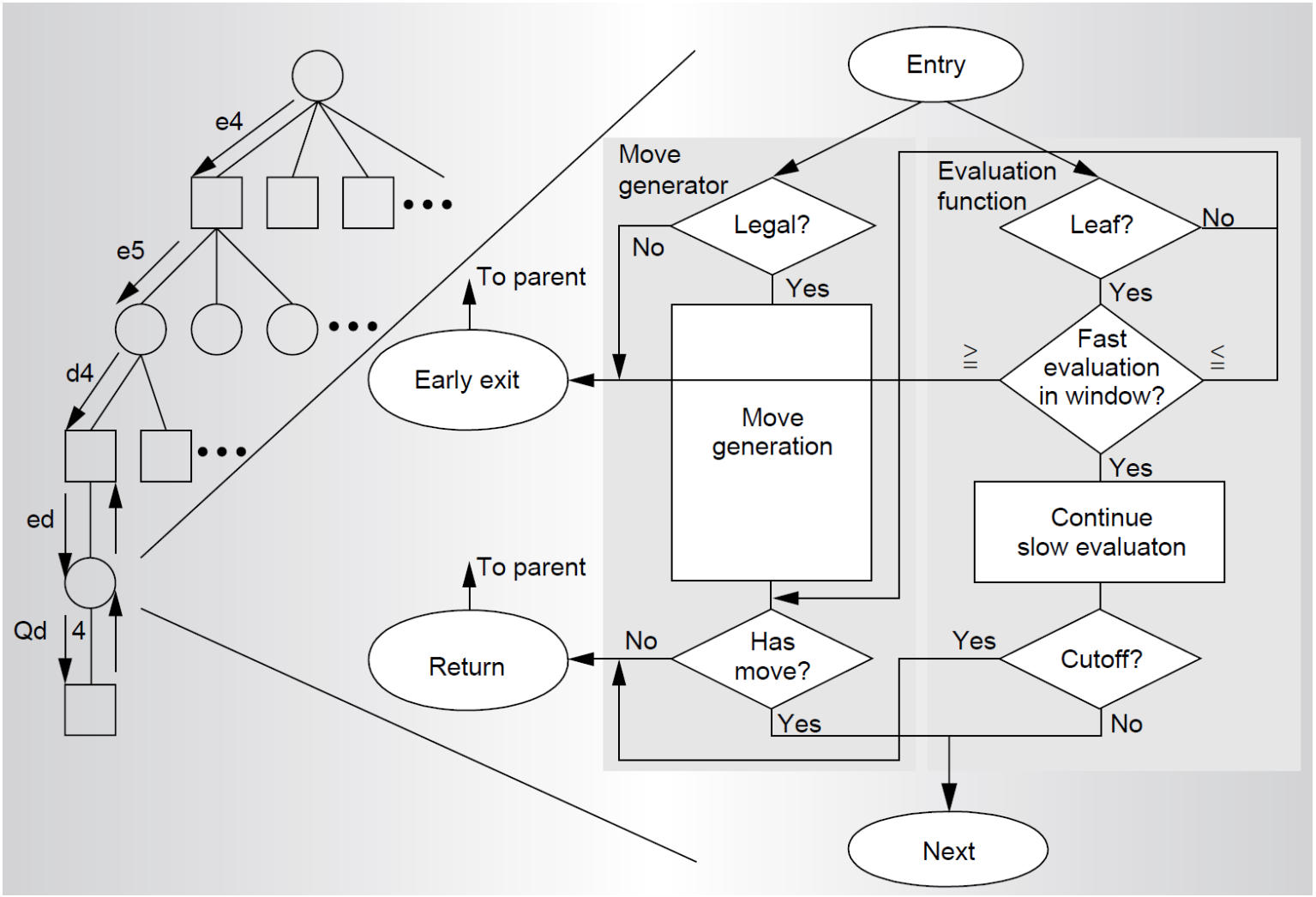
IBM Deep Blue



Block diagram of the chess chip

Feng-hsiung Hsu, 1999

IBM Deep Blue



A chess chip's basic search algorithm: search tree (left), flow chart (right).

IBM Deep Blue

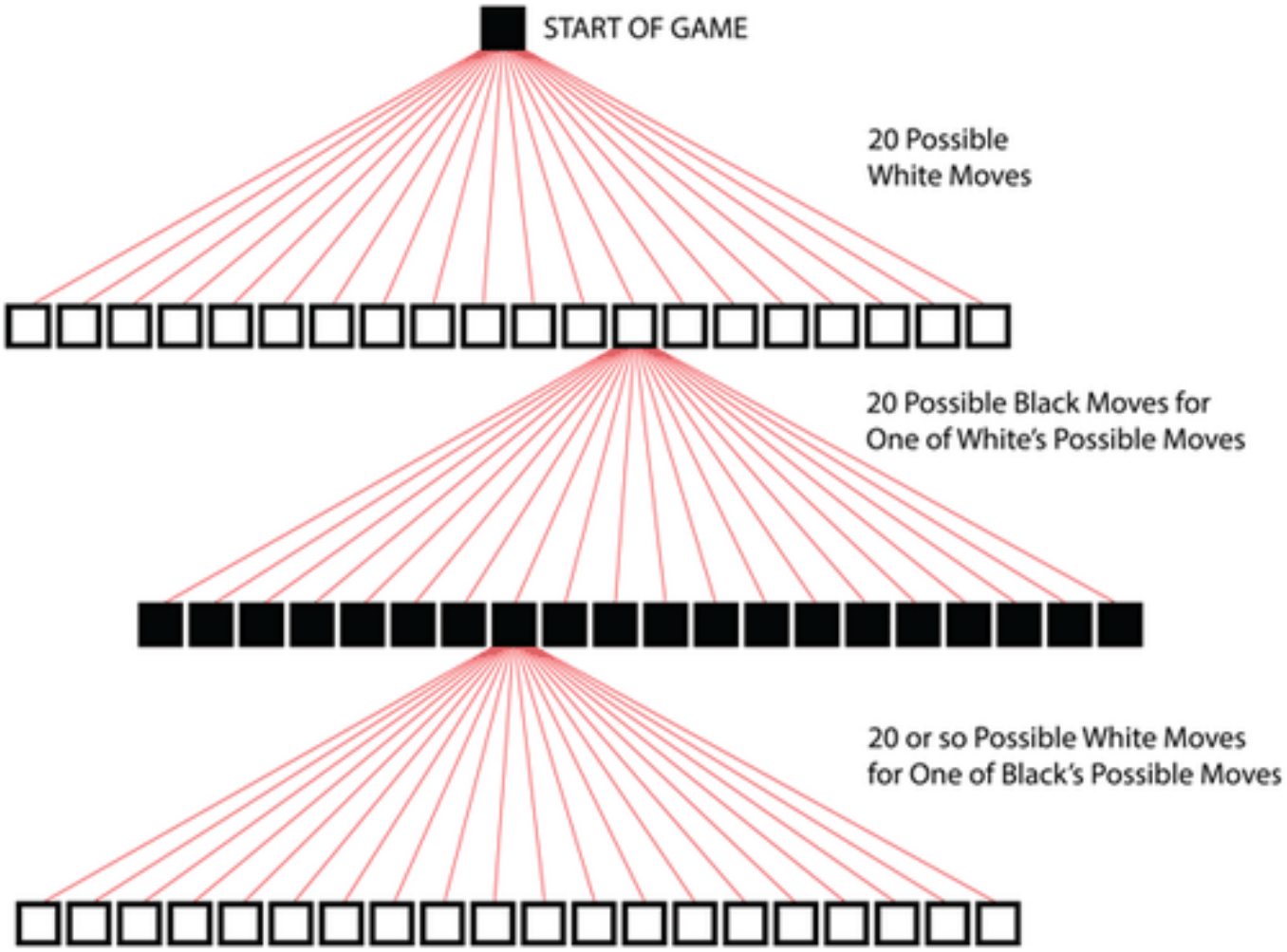


Image credit: Laura Guerin

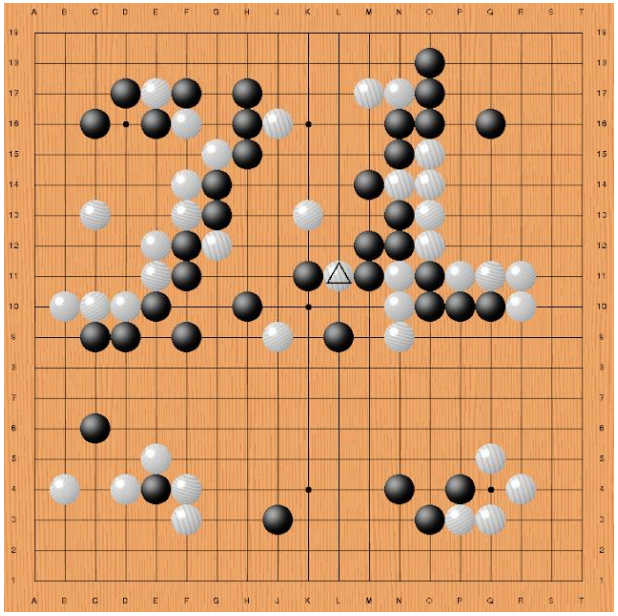
IBM Deep Blue

- For all the possible chess board positions, you need to evaluate 10^120 possible moves.
  - It takes 10^108 years if the computer can calculate one billion moves per second.
- Deep Blue:
  - search 40 billion positions for each move.
  - Search was distributed
  - Evaluate 200 million moves per second.

Google AlphaGo

- AlphaGo is a computer program that plays the board game Go. It was developed by DeepMind, which was later acquired by Google.

- The Game of Go
  - Played on a square grid called a board, usually 19 x 19
  - Stones – black and white – are placed alternatively
  - Points awarded for surrounding empty space



- Complexity of Go
  - Possible number of sequences ≈ 250^150
  - 10^100 times more complex than chess
  - Viewed as an unsolved "grand challenge" for AI

Google AlphaGo

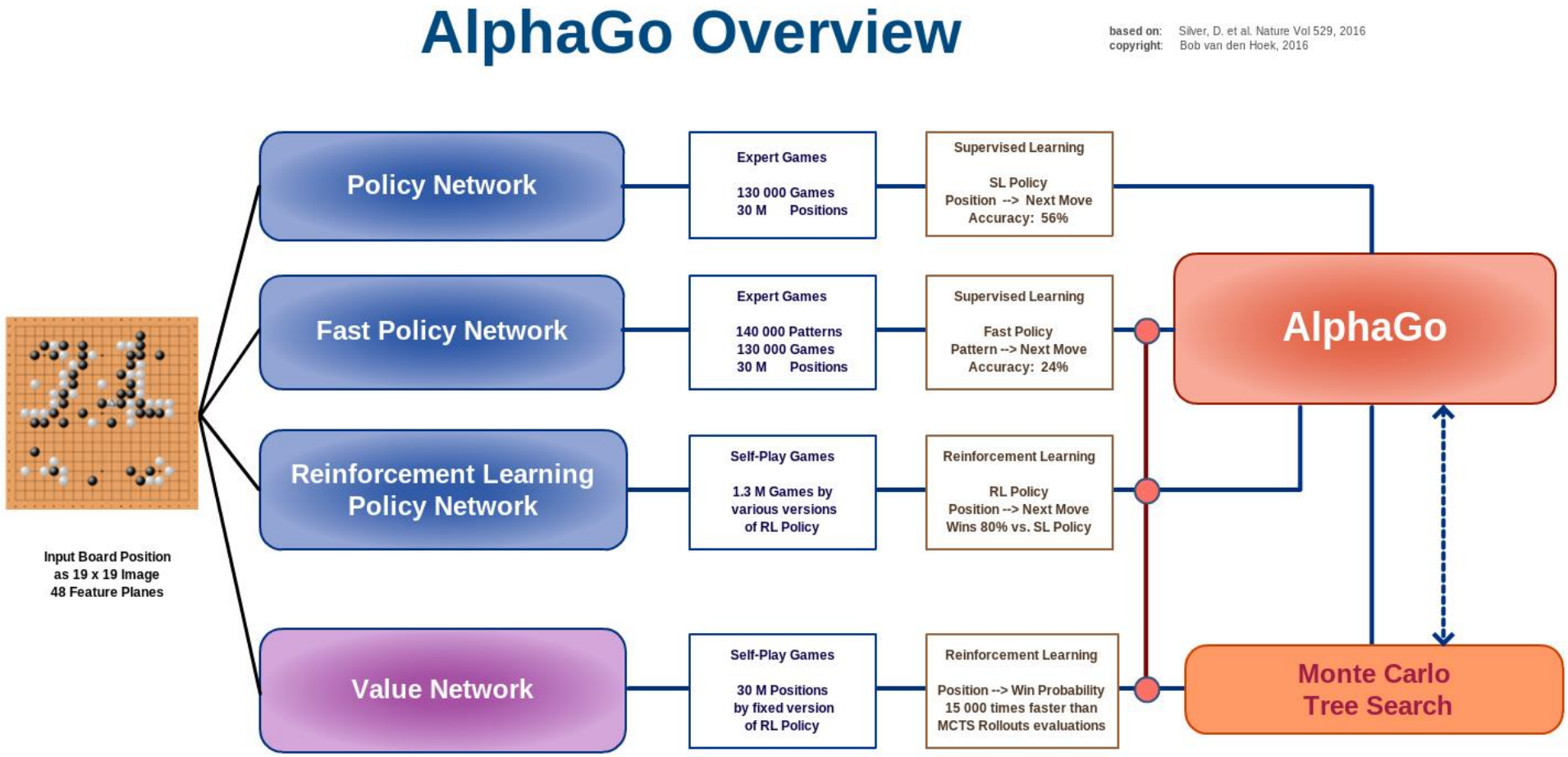
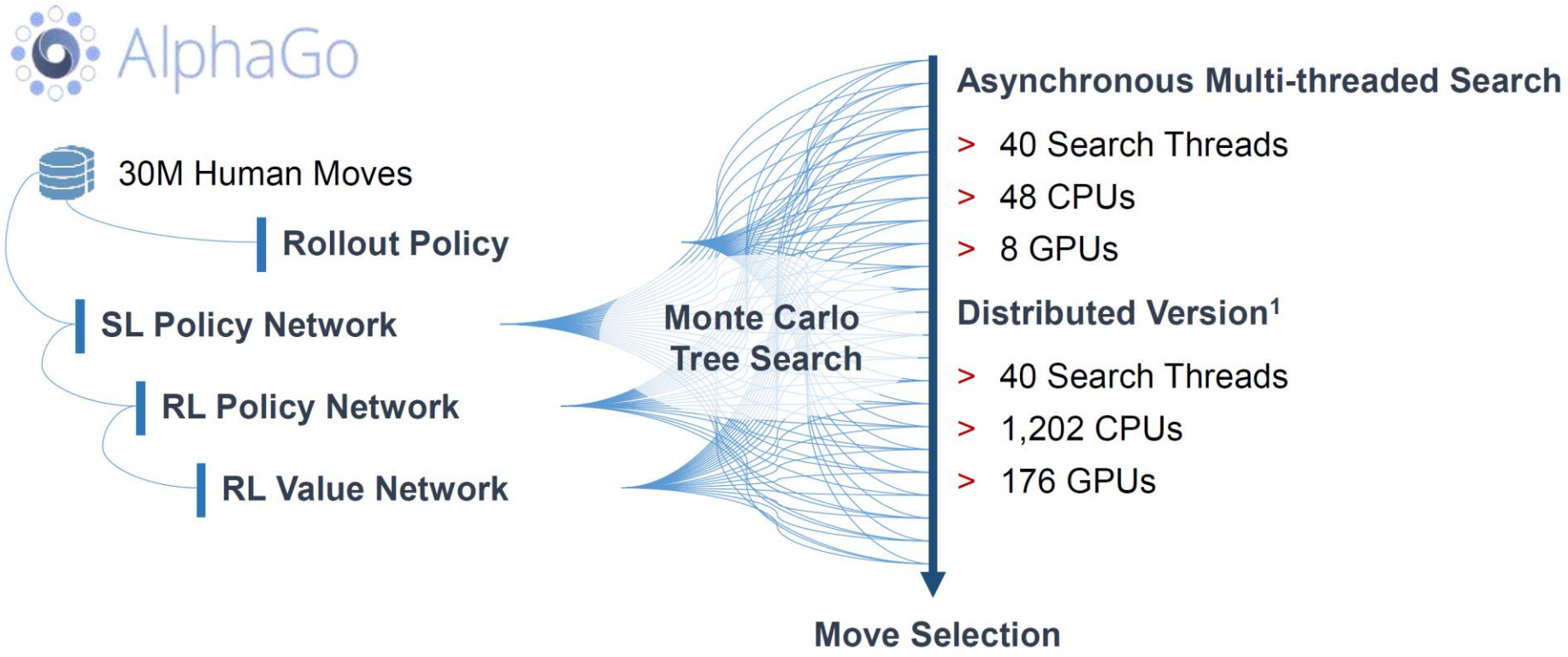


Image credit

Google AlphaGo



1. Used against Fan Hui; 1,920 CPUs and 280 GPUs against Lee
http://www.economist.com/news/science-and-technology/21694540-win-or-lose-best-five-battle-contest-another-milestone

Source: M.Moon, UoT

Alpha Go zero

https://www.youtube.com/watch?v=tXlM99xPQC8

ImageNet

- ImageNet is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images. Currently ImageNet an average of over five hundred images per node.

  http://www.image-net.org/
- ImageNet accelerates the development of computer vision techniques (supervised especially)
- ImageNet becomes a useful resource for researchers, educators, students

ImageNet

https://www.youtube.com/watch?v=40riCqvRoMs&t=21s

AI Landscape