

# Eine Architektur für Programmsynthese aus natürlicher Sprache

Mathias Landhäußer

IPD Tichy, Fakultät für Informatik

Es gibt viel Forschung zum Thema  
„Software-Entwicklung...

**effizienter,**

**schneller, [...]**

**einfacher**

**machen“.**

Nur etwa **0,3 Prozent**  
der Weltbevölkerung  
können überhaupt  
**Software entwickeln.**<sup>1,2</sup>

- 1) "There are 18.5 million software developers in the world - of which 11 million are professionals and 7.5 million are hobbyists."
- 2) <http://www.techrepublic.com/blog/european-technology/there-are-185-million-software-developers-in-the-world-but-which-country-has-the-most/>

Auf der anderen Seite gibt es etwa  
**2 Mrd. PCs<sup>3</sup>** und  
**7 Mrd. Tablets**  
und Smartphones<sup>4</sup>.

3) <http://www.worldometers.info/computers/>

4) <http://www.independent.co.uk/life-style/gadgets-and-tech/news/there-are-officially-more-mobile-devices-than-people-in-the-world-9780518.html>

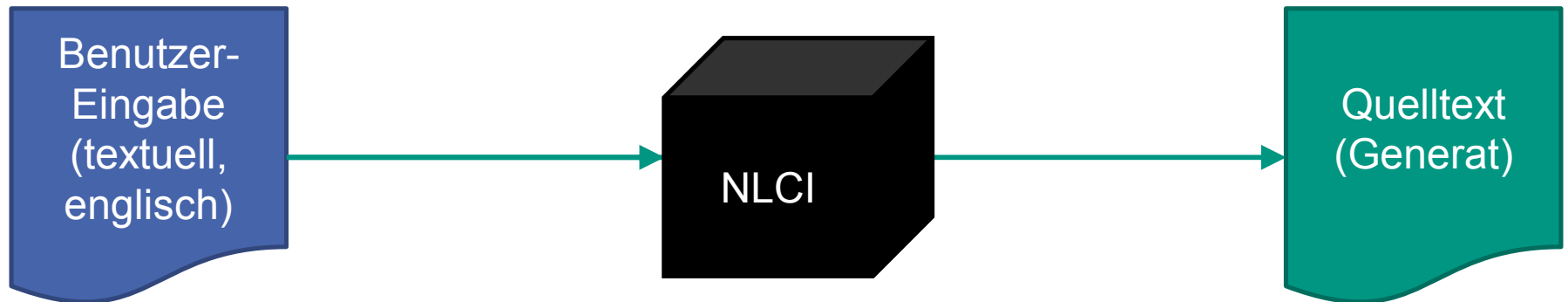
Wie/wann können die übrigen **99,7 Prozent**  
programmieren?



“The only way a person can  
**truly concentrate on his problem and solve it** [. . .]  
are if he is able to **communicate directly** with the computer  
**without having to learn** some specialized intermediate language.”

Jean E. Sammet, 1966

# Ziel dieser Arbeit

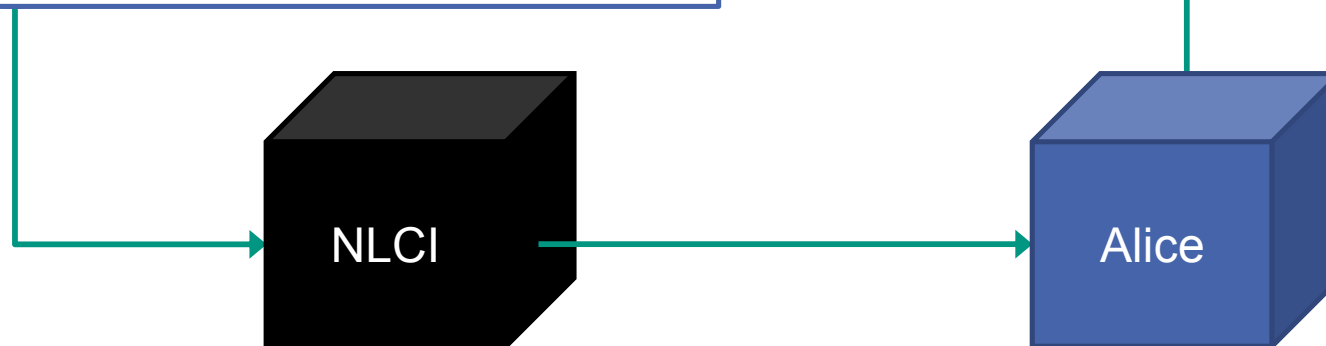


**Natural Language Command Interpreter**

# Einführendes Beispiel

## Erzeugen von Animationen mit CMU Alice

The bunny says, "Hello". Alice answers to the bunny. Alice says, "???". The bunny **jumps three times** and then says, "Come over here". Alice **shakes her head**. The bunny says, "Come on! What time is it?". Alice goes to the bunny. **While she goes**, she says, "Ohh" and after a little while, "Okay". A cat with a broad grin appears on the tree. The cat **turns its head** to the bunny and says, ":-D". Alice and the bunny turn their heads to the cat **at the same time**. The bunny turns its whole body to face the black hole. While it turns, it says, "Ahhhhhhh!". The bunny jumps to the hole and then jumps into the hole. **While the bunny jumps**, Alice turns to the bunny and says, "Hey, Wait".





# Verwandte Arbeiten & Stand der Kunst

**“I suspect that machines to be programmed in our native tongues [...] are as damned difficult to make as they would be to use.”**

[Edsger W. Dijkstra]

1960s

„Impossible”

„no need“

[Dijkstra 1963]

[Dijkstra 1964]

[Hill 1972]

# Verwandte Arbeiten & Stand der Kunst

"Choose a row in the matrix."

"Put the average of the first four entries in that row into its last entry."

```
select($A)
```

```
A30 := avg(A1:A4)
```

[Ballard1979]

● 1979

Natural  
Language  
Computer  
[Ballard1979]

1960s

„Impossible“

„no need“

[Dijkstra1963]

[Dijkstra1964]

[Hill1972]

# Verwandte Arbeiten & Stand der Kunst

“There is a bar with a bartender who makes drinks.”

```
class bar:
    the_bartender = bartender()
class bartender:
    def make(drink): pass
```

[Liu2005]

● 1979  
 Natural  
 Language  
 Computer  
 [Ballard1979]

● 2000er  
 Programm-  
 rümpfe aus  
 Texten  
 [Liu2005]  
 Studien zur  
 Ausdrucks-  
 weise von  
 Laien  
 [Pane2001,2002]

1960s  
 „Impossible”  
 „no need“  
 [Dijkstra1963]  
 [Dijkstra1964]  
 [Hill1972]

# Verwandte Arbeiten & Stand der Kunst



[Ferruci2010]

1960s  
„Impossible“  
„no need“  
[Dijkstra1963]  
[Dijkstra1964]  
[Hill1972]

● 1979  
Natural  
Language  
Computer  
[Ballard1979]

● 2000er  
Programm-  
rumpfe aus  
Texten  
[Liu2005]  
Studien zur  
Ausdrucks-  
weise von  
Laien  
[Pane2001,2002]

● 2010er  
IBM Watson  
[Ferruci2010]  
SmartSynth  
[Le2013]  
Apples Siri  
[Bellegarda2014]

# Verwandte Arbeiten & Stand der Kunst

## Stand der Kunst

- Eingeschränkte Formulierungen/Wortwahl  
→ **Benutzbarkeit?**
- Domänenabhängig:  
→ **Übertragbarkeit?**

1960s  
„Impossible”  
„no need“  
[Dijkstra1963]  
[Dijkstra1964]  
[Hill1972]

● 1979  
Natural  
Language  
Computer  
[Ballard1979]

● 2000er  
Programm-  
rumpfe aus  
Texten  
[Liu2005]  
Studien zur  
Ausdrucks-  
weise von  
Laien  
[Pane2001,2002]

● 2010er  
IBM Watson  
[Ferruci2010]  
SmartSynth  
[Le2013]  
Apples Siri  
[Bellegarda2014]

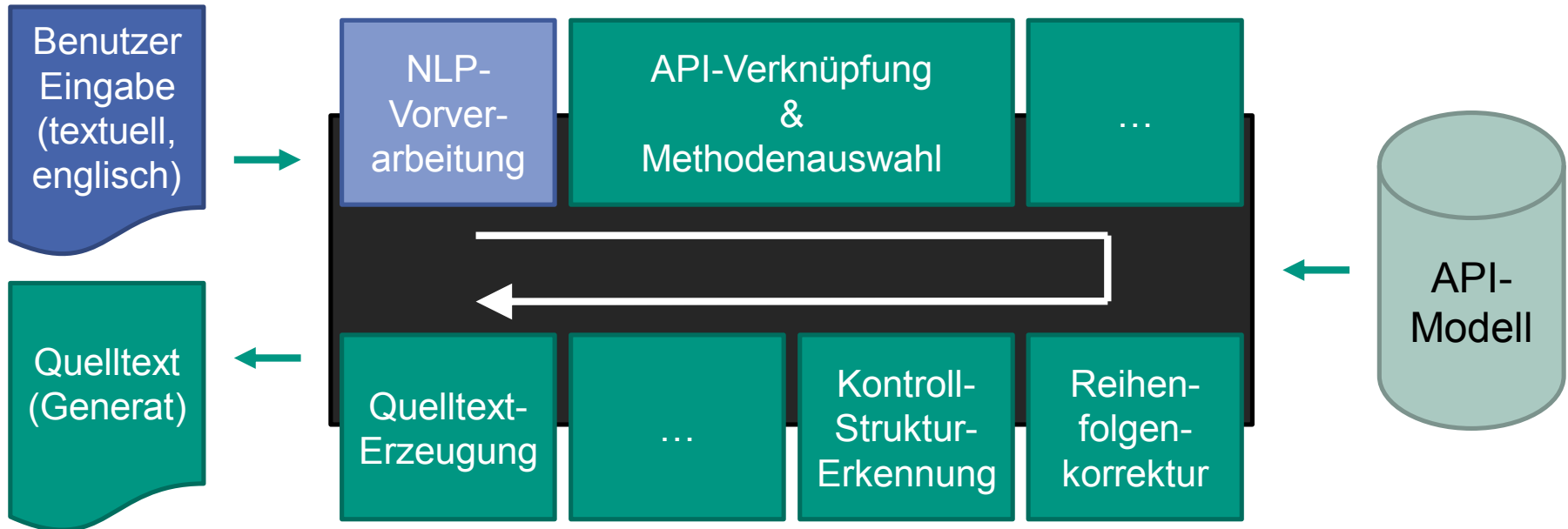
● Programmieren  
in natürlicher  
Sprache  
[?]

# Überblick über die Architektur



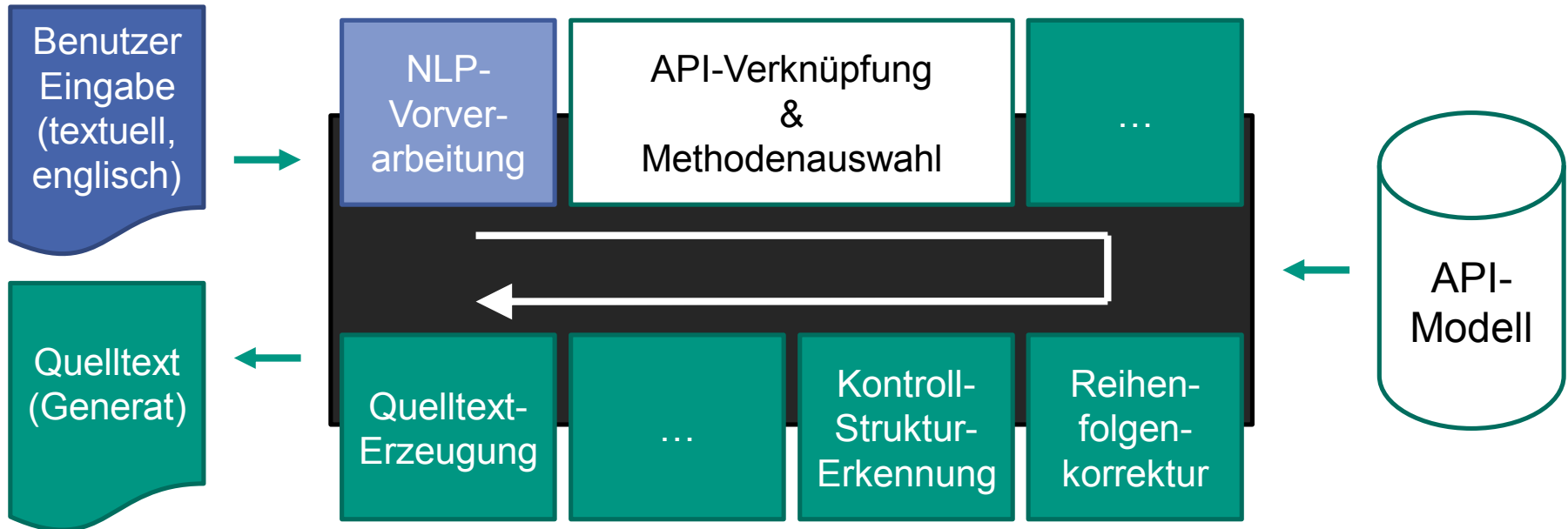
- Modulare Architektur zur Übersetzung von Text nach Quelltext
  - Entkopplung von Textanalyse und Wissen über die API
  - API ist Teil der Konfiguration von NLCI
  - API-Modell kann automatisch erzeugt werden und wird automatisch aufbereitet (Bezeichner, Synonyme, etc.)

# Überblick über die Architektur



- Modulare Architektur zur Übersetzung von Text nach Quelltext
  - Entkopplung von Textanalyse und Wissen über die API
  - API ist Teil der Konfiguration von NLCI
  - API-Modell kann automatisch erzeugt werden und wird automatisch aufbereitet (Bezeichner, Synonyme, etc.)

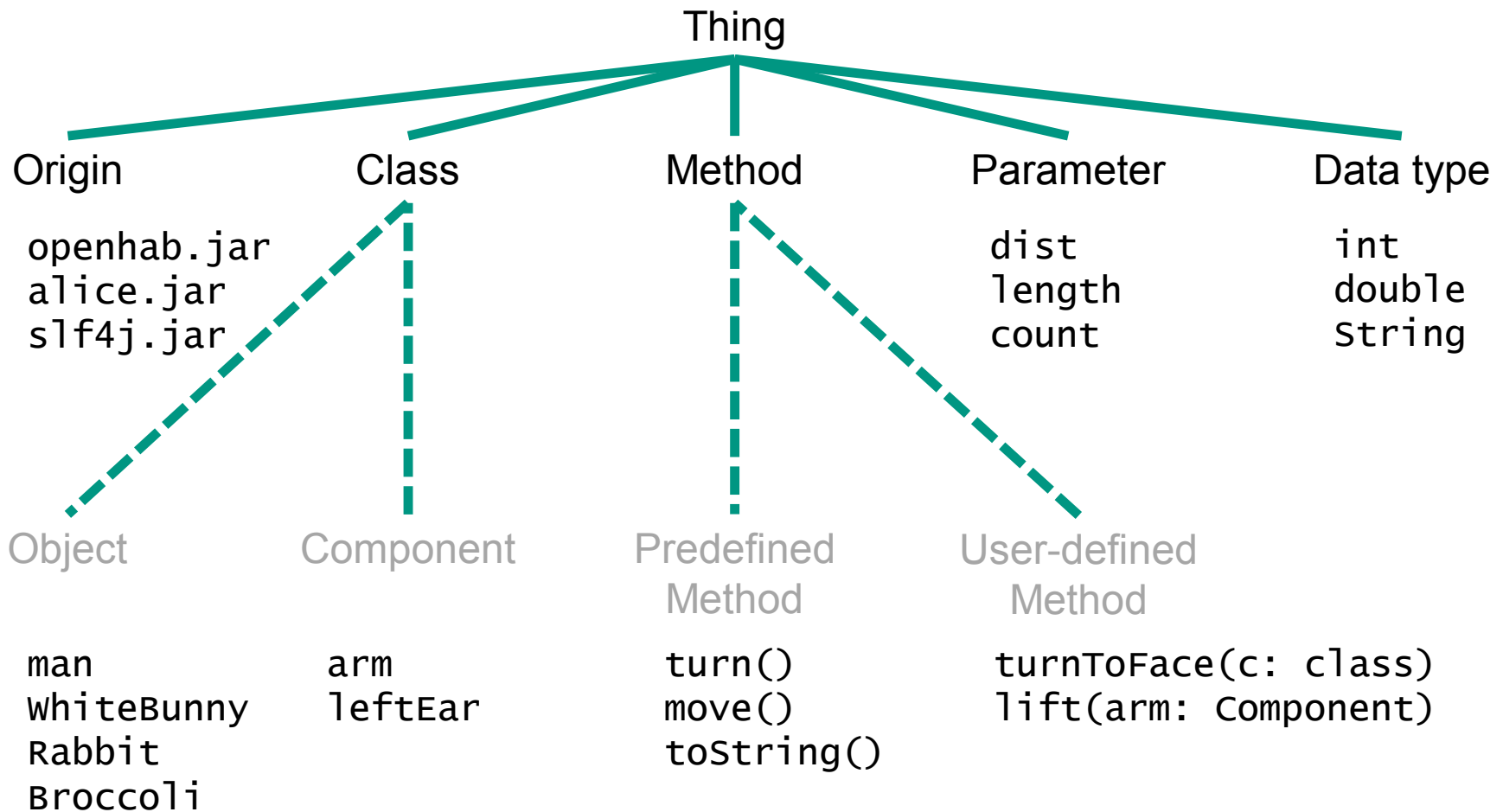
# Überblick über die Architektur



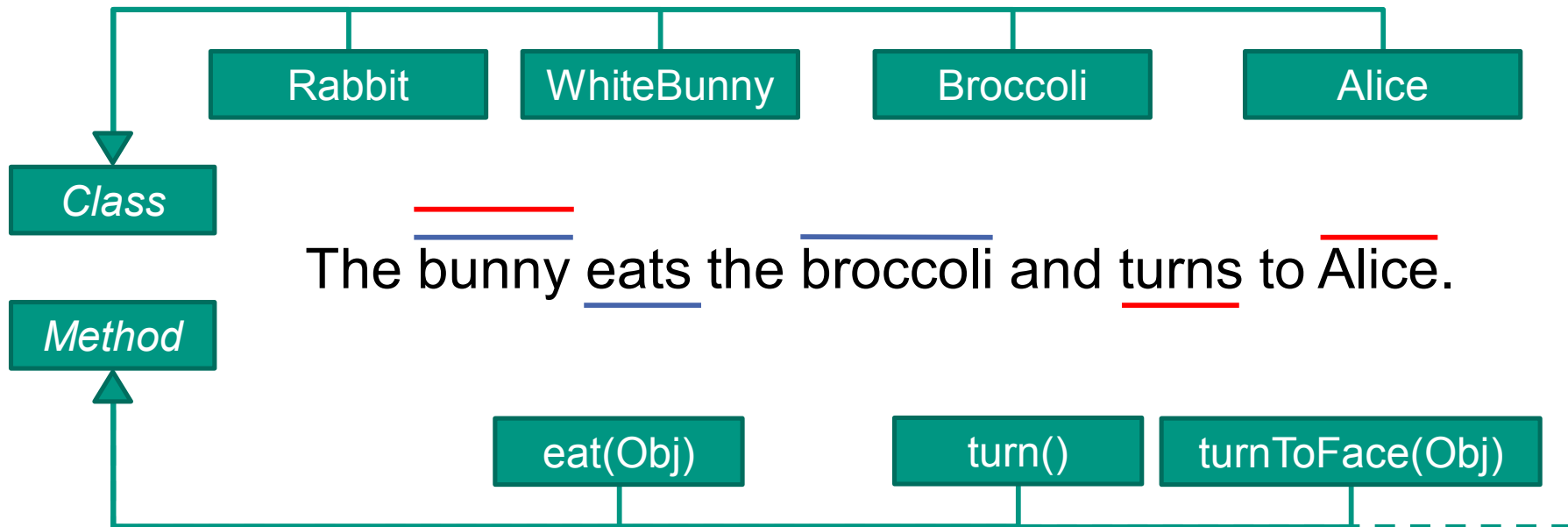
- Modulare Architektur zur Übersetzung von Text nach Quelltext
  - Entkopplung von Textanalyse und Wissen über die API
  - API ist Teil der Konfiguration von NLCI
  - API-Modell kann automatisch erzeugt werden und wird automatisch aufbereitet (Bezeichner, Synonyme, etc.)



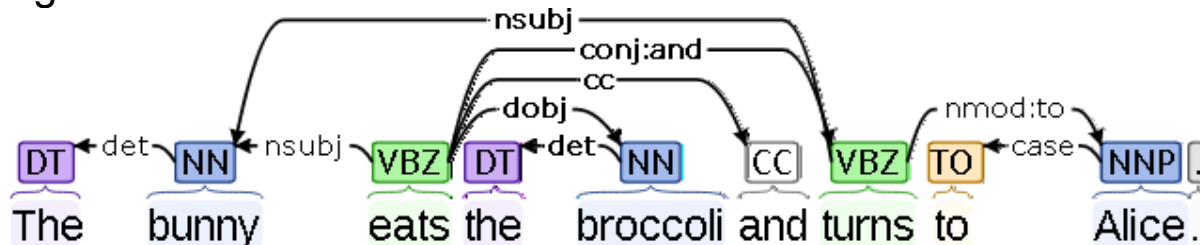
# Verknüpfung der Textelemente mit der API: API-Darstellung im Modell



# Verknüpfung der Textelemente mit der API: Satzanalyse und API-Suche

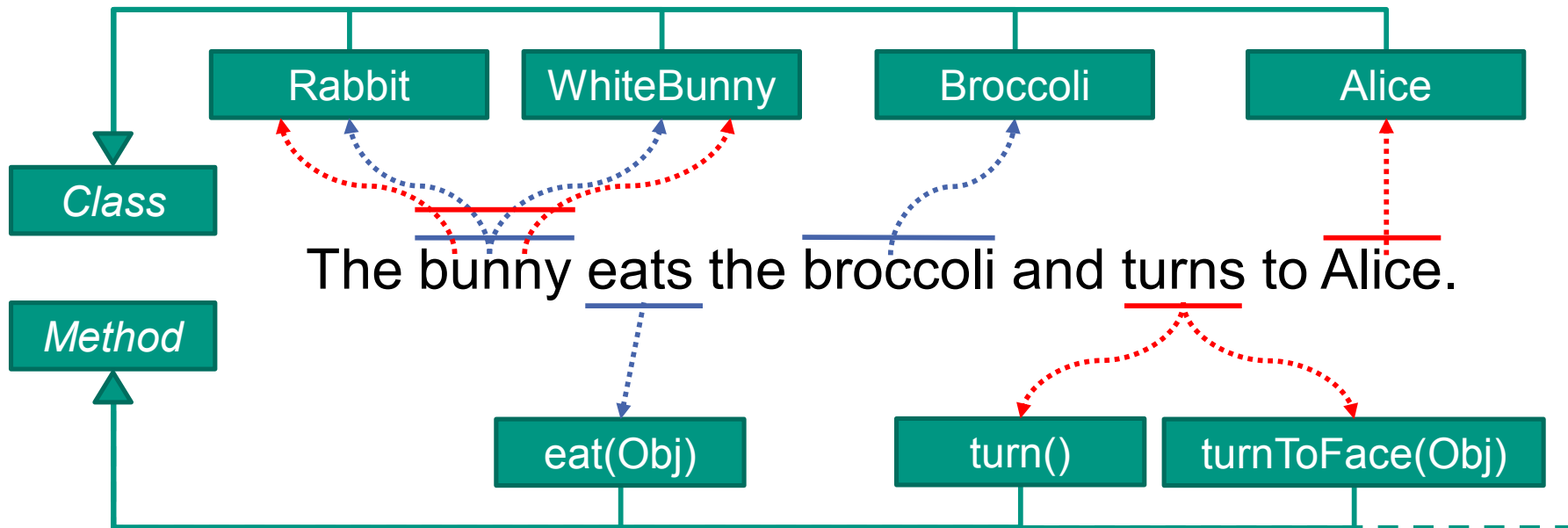


## ■ Vorverarbeitung mit Stanford CoreNLP



- Textanalyse betrachtet Aktiv und Passiv, Imperative, Adjektive, Nebensätze, Konjunktionen, Appositionen etc.

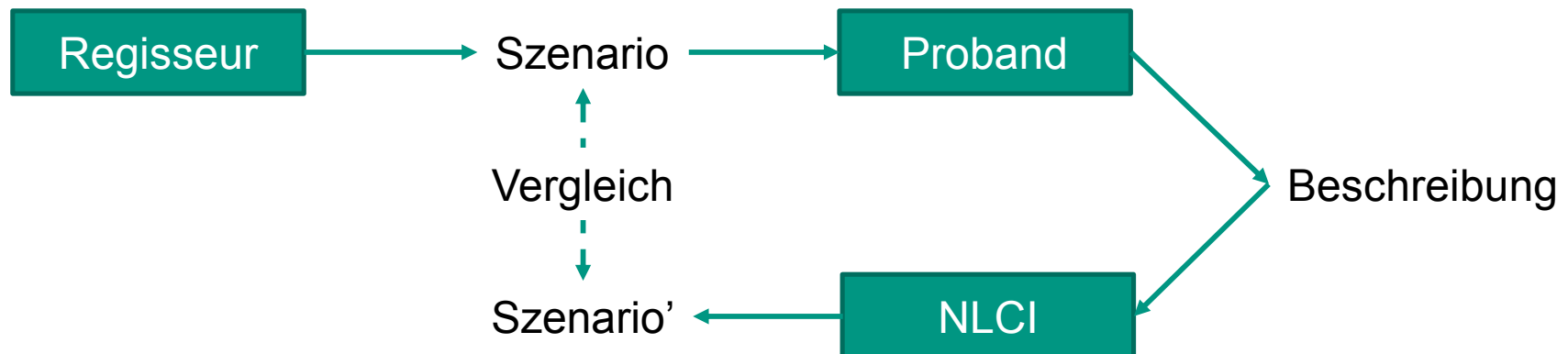
# Verknüpfung der Textelemente mit der API: Satzanalyse und API-Suche



- Kandidaten für „bunny eats broccoli“
  - ~~whiteBunny.eat(Broccoli)~~, Rabbit.eat(Broccoli)
- Kandidaten für „bunny turns to Alice“
  - Rabbit.turnToFace(Alice), ~~Rabbit.turn()~~, ~~whiteBunny.turn()~~

# Fallstudien und Evaluation: Überblick und Vorgehen

- Heimautomation mit openHAB
  - Texte eher im Imperativ „schalte das Licht in der Küche ein“...
  - 114 Klassen mit insgesamt 9 Methoden
  - Kleinere Machbarkeitsstudie
- 3D-Animationen mit Alice
  - Texte sind beschreibende Prosa
  - 914 Klassen mit insgesamt 393 Methoden
  - Umfangreiche Auswertung von 103 Eingabetexten



# Evaluation: Konstruktion von Methodenaufrufen

## CMU Alice & openHAB

### ■ Korpora

|           | Texte | Sätze | Wörter | API-Aufrufe |
|-----------|-------|-------|--------|-------------|
| CMU Alice | 50    | 703   | 6764   | 570         |
| openHAB   | 5     | 15    | 149    | 20          |

### ■ Ergebnisse

|           | Metric [%]     | 0 Param. | 1 Param. | 2 Param. | Gesamt |
|-----------|----------------|----------|----------|----------|--------|
| CMU Alice | Präzision      | 85,7     | 86,1     | 100,0    | 86,5   |
|           | Ausbeute       | 74,5     | 70,9     | 54,1     | 71,5   |
|           | F <sub>1</sub> | 79,7     | 77,8     | 70,2     | 78,8   |
| openHAB   | Präzision      | 73,7     |          |          | 73,7   |
|           | Ausbeute       | 70,0     |          |          | 70,0   |
|           | F <sub>1</sub> | 71,8     |          |          | 71,8   |

# Zusammenfassung & Ausblick

- Die vorgestellte Architektur
  - erlaubt Programmsynthese aus englischem Text und
  - erleichtert die Erschließung von neuen Domänen (oder APIs).
- Die Evaluation des Prototyps zeigt, dass man
  - keine Einschränkung der Eingabesprache machen muss,
  - keine chronologisch korrekte Beschreibung fordern muss und
  - Kontrollstrukturen aus der Eingabe synthetisieren kann.
- Möchte man NLCI weiterentwickeln, dann sollte man
  - Fehler der NLP-Vorverarbeitung verringern oder ausgleichen,
  - mehr Sprachverständnis erreichen (z. B. Korreferenzen) und
  - weitere API-Eigenschaften nutzen (z. B. Vor- und Nachbedingungen).

**Vielen Dank!**

# Referenzen

- [Ballard1979] Ballard, Bruce W., and Alan W. Biermann. „Programming in Natural Language: NLC as a Prototype.“ In Proceedings of the 1979 Annual Conference, 228–37. ACM '79. New York, NY, USA: ACM, 1979. [doi:10.1145/800177.810072](https://doi.org/10.1145/800177.810072).
- [Bellegarda2014] Bellegarda, Jerome R. “Spoken Language Understanding for Natural Interaction: The Siri Experience.” In Natural Interaction with Robots, Knowbots and Smartphones, edited by Joseph Mariani, Sophie Rosset, Martine Garnier-Rizet, and Laurence Devillers, 3–14. Springer New York, 2014. [doi:10.1007/978-1-4614-8280-2\\_1](https://doi.org/10.1007/978-1-4614-8280-2_1).
- [Dijkstra1963] Dijkstra, Edsger W. „On the Design of Machine Independent Programming Languages.“ Annual Review in Automatic Programming, Annual Review in Automatic Programming, 3 (1963): 27–42. [doi:10.1016/S0066-4138\(63\)80003-8](https://doi.org/10.1016/S0066-4138(63)80003-8).
- [Dijkstra1964] ———. „Some Comments on the Aims of MIRFAC.“ Commun. ACM 7, no. 3 (March 1964): 190. [doi:10.1145/363958.364002](https://doi.org/10.1145/363958.364002).
- [Ferruci2010] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, N. Schlaefer, and C. Welty, “Building Watson: An Overview of the DeepQA Project,” *AI Magazine*, vol. 31, no. 3, pp. 59–79, 2010.
- [Hill1972] Hill, I. D. „Wouldn't It Be Nice If We Could Write Computer Programs in Ordinary English - or Would It?“ Computer Bulletin 16, no. 6 (1972): 306–12.
- [Kiesel2016] Kiesel, Viktor. „Optimieren von Wortartmarkiererergebnissen.“ Bachelorarbeit, Karlsruher Institut für Technologie (KIT) – IPD Tichy, April 2016. [https://code.ipd.kit.edu/weigelt/parse/wikis/Theses/kiesel\\_ba](https://code.ipd.kit.edu/weigelt/parse/wikis/Theses/kiesel_ba).
- [Le2013] Le, Vu, Sumit Gulwani, and Zhendong Su. „Smartsynth: Synthesizing Smartphone Automation Scripts from Natural Language.“ In MobSys'13, 2:5, 2013.
- [Lieberman2006] Lieberman, Henry, and Hugo Liu. „Feasibility Studies for Programming in Natural Language.“ In End User Development, edited by Henry Lieberman, Fabio Paternò, and Volker Wulf, 459–73. Human-Computer Interaction Series 9. Springer Netherlands, 2006. [http://link.springer.com/chapter/10.1007/1-4020-5386-X\\_20](http://link.springer.com/chapter/10.1007/1-4020-5386-X_20).
- [Pane2000] Pane, John F., Brad A. Myers, and L.B. Miller. „Using HCI Techniques to Design a More Usable Programming System.“ In IEEE 2002 Symposia on Human Centric Computing Languages and Environments, 2002. Proceedings, 198–206, 2002. [doi:10.1109/HCC.2002.1046372](https://doi.org/10.1109/HCC.2002.1046372).
- [Pane2001] Pane, John F., Chotirat Ratanamahatana, and Brad A. Myers. „Studying the Language and Structure in Non-Programmers' Solutions to Programming Problems.“ International Journal of Human-Computer Studies 54, no. 2 (2001): 237–64. [doi:10.1006/ijhc.2000.0410](https://doi.org/10.1006/ijhc.2000.0410).
- [Price2000] Price, David, Ellen Riloff, Joseph Zachary, and Brandon Harvey. „NaturalJava: A Natural Language Interface for Programming in Java.“ In Proceedings of the 5th International Conference on Intelligent User Interfaces, 207–11. IUI '00. New Orleans, Louisiana, USA: ACM, 2000. [doi:10.1145/325737.325845](https://doi.org/10.1145/325737.325845).



# Bildquellen

- Abhängigkeitsgraph: [nlp.stanford.edu](http://nlp.stanford.edu)
- Hello Barbie: [www.amazon.com](http://www.amazon.com)
- IBM Watson: [www.watson2016.com](http://www.watson2016.com)
- openHAB: [www.openhab.org](http://www.openhab.org)
- Siri: [www.applemagazine.com](http://www.applemagazine.com)