

# Nearest Neighbor Queries

---

In a loosely coupled  
set of data sources

## **SI Mobiles**

Course of lectures of Ahmed LBATH

Lecture of Pascal BIHLER ([pb@bi-on.de](mailto:pb@bi-on.de))



**Universität Stuttgart**

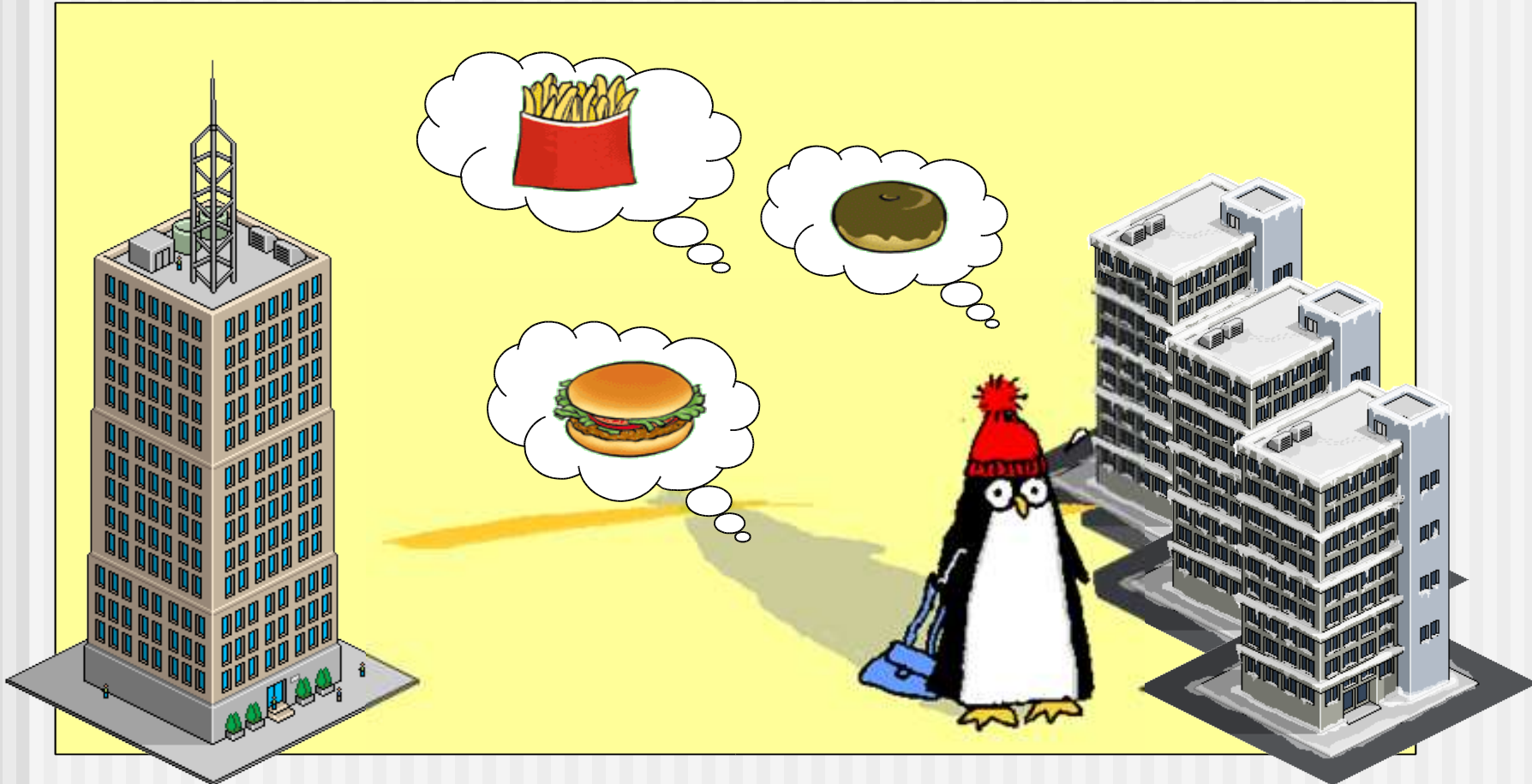
# Summary

---

- Introduction
- Objectives
- Nearest Neighbor Queries in general
- Federated Nearest Neighbor Queries
- Experimental results
- Criticism & Conclusion
- Questions/Discussion

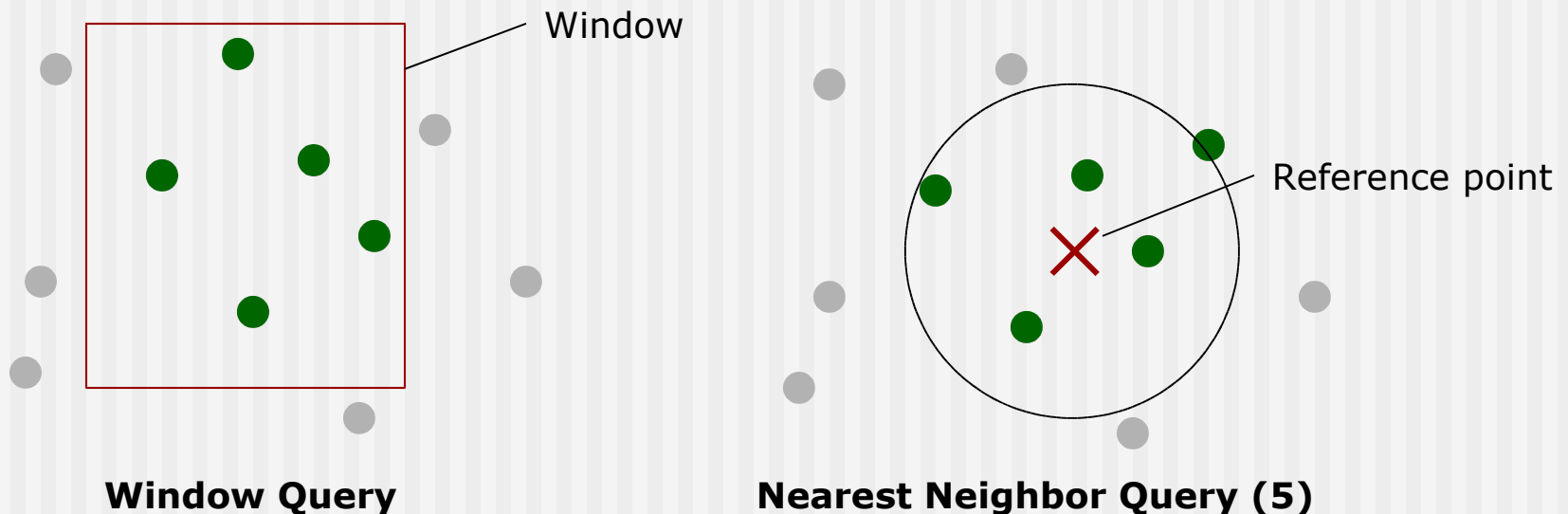


# Classical example



# Spatial Data Queries

- Two kinds of queries:
  - Window Queries
  - Nearest Neighbor (NN) Queries



# Objectives

---

- Metrics to estimate NN queries:
  - Response time
  - Effort
  - Iterations
  - Cost profit ratio
  
- Use degree of parallelism to balance minimal response time with minimal resource consumption

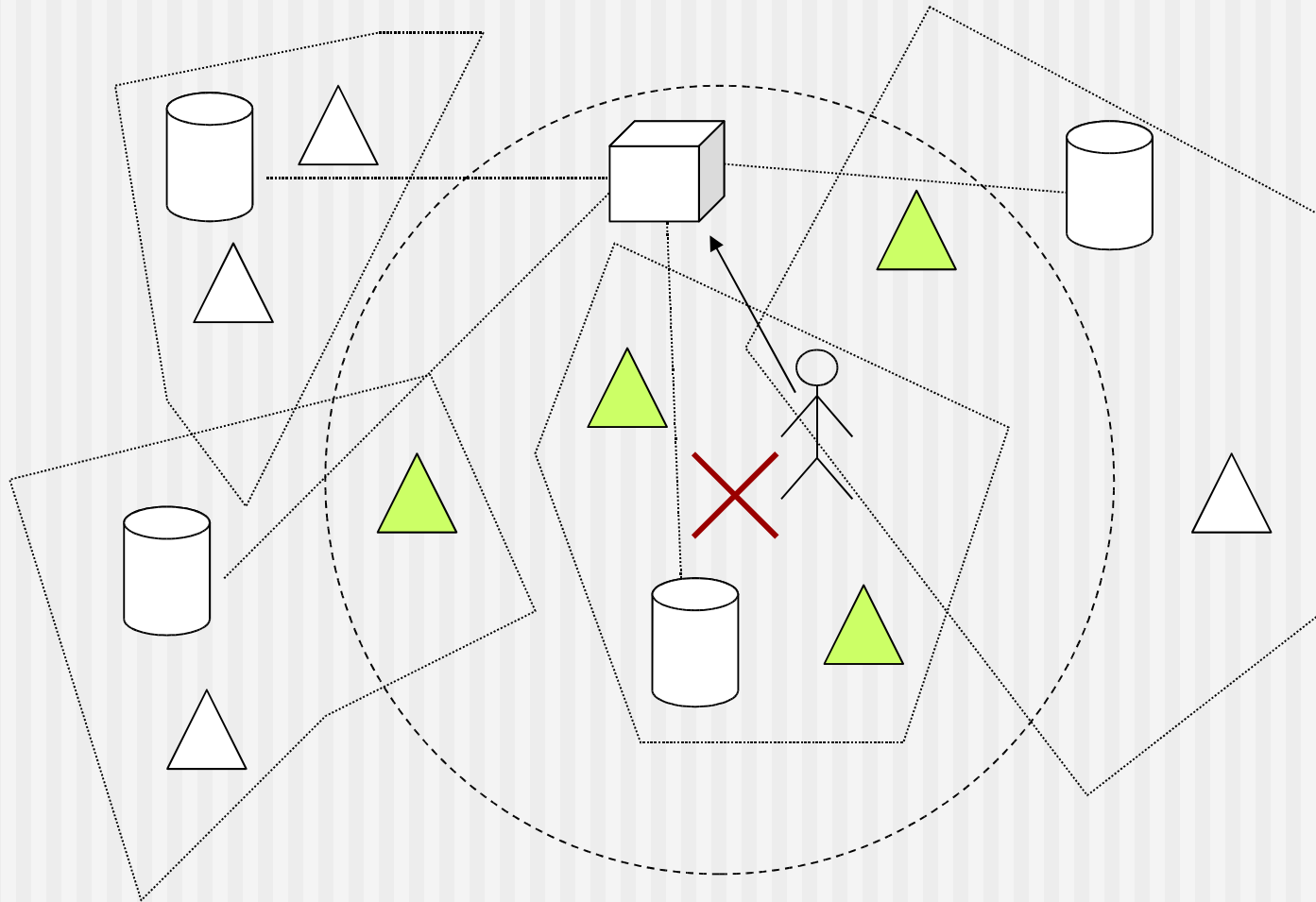


# Nearest Neighbor Queries



- Relatively easy if distribution is known
- More difficult if exact object count and density statistics are unknown
- Four individual attempts:
  - Local Nearest Neighbor Queries
  - Emulated Nearest Neighbor Queries
  - Distributed and Parallel NN Queries
  - Federated Nearest Neighbor Queries

# Problem Structure



# Simple Query Approaches



- Local Nearest Neighbor Queries
  - Query a local database
  - Restricted to a local processing
  - Not extensible to remote data sources
- Emulated Nearest Neighbor Queries
  - Use Window Queries to emulate NN query
  - Improvement: Use n-corner polygons ( $n > 4$ )

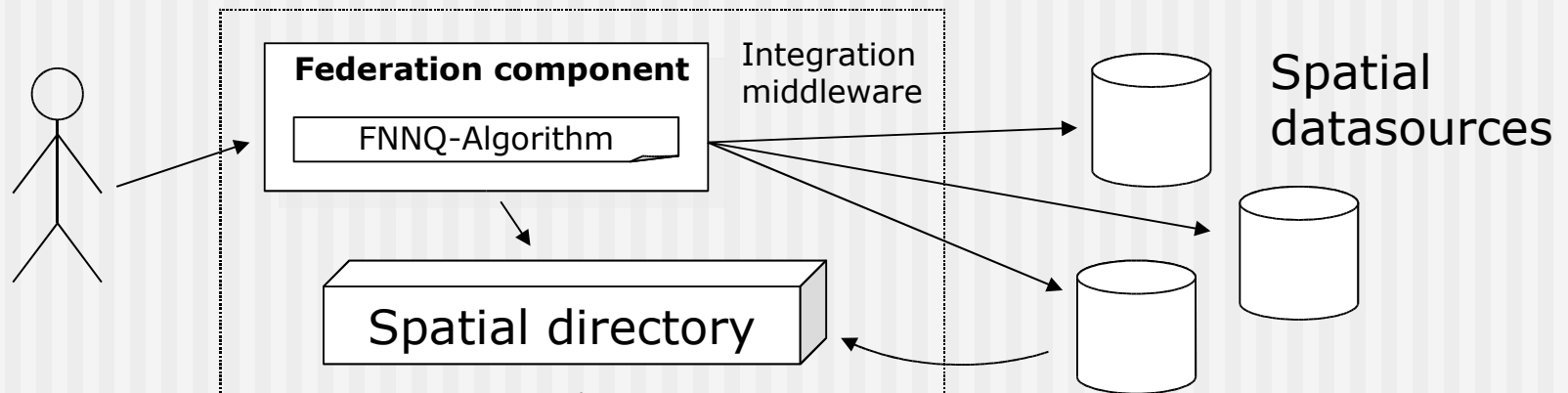
# Distributed and Parallel NN Queries



- Mostly the data is cleverly partitionated
  - High parallel data fetch possible
  - Query all sources the same time
  - Data structured as distributed R-tree
- In our scenario:
  - Partitions may overlap (Not influenced)
  - Query as few as possible
  - Adapting number of threads dynamically
  - Just two levels (root & source databases)
    - No benefits from optimization of R-tree algorithms

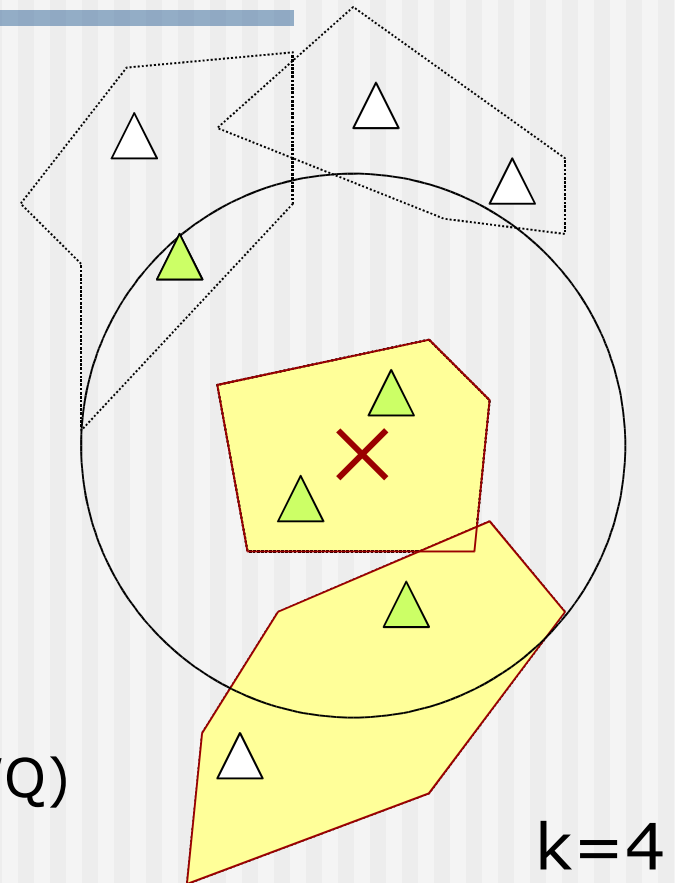
# Federated Nearest Neighbor Queries

- Insufficiencies of existent algorithms
  - Rely on object count
  - Access all data sources at the same time
  - Target a different system architecture
- Federated Nearest Neighbor Queries



# FNNQ Algorithm

- Initial Phase
- Expansion Phase
- Convergence Phase
- Datasource Query Methods
  - Remote NN Queries (RNNQ)
  - Remote Window Queries (RWQ)
  - Querying in parallel
  - Querying incrementally



# Experiments

---

- Experimental parameters:
  - Environment size: Germany
  - 10,000 databases
  - 1,000,000 objects (uniform. distr.)
  - $k$  varying (1 -> 1024)
  - #threads varying (1 -> 10,000)
  
- Communication delays
  - Raise significantly the overall response time
  - Therefore no incrementally data source queries estimated

# Experiment Results

---

- Overall speedup up to 3.5 to other algorithms known from literature
- The more threads are used, the lower the response time is (except for the case of 100% cov.)
- For typical scenarios ( $k=10$ ), easy approaches are nearly as good as more difficult one (using global object density, getting better results for bigger  $k$ )

# Criticism

---

- Problem discussed for 2D-applications only (“can be easily generalized”)
- RNNQ and RWQ already need server lists ordered by distance (how to assure consistence in dynamic surroundings?)
- Experiments may not be significant
  - Simulated server environment
  - No freely available real world dataset to test
  - Uniform distribution realistic?

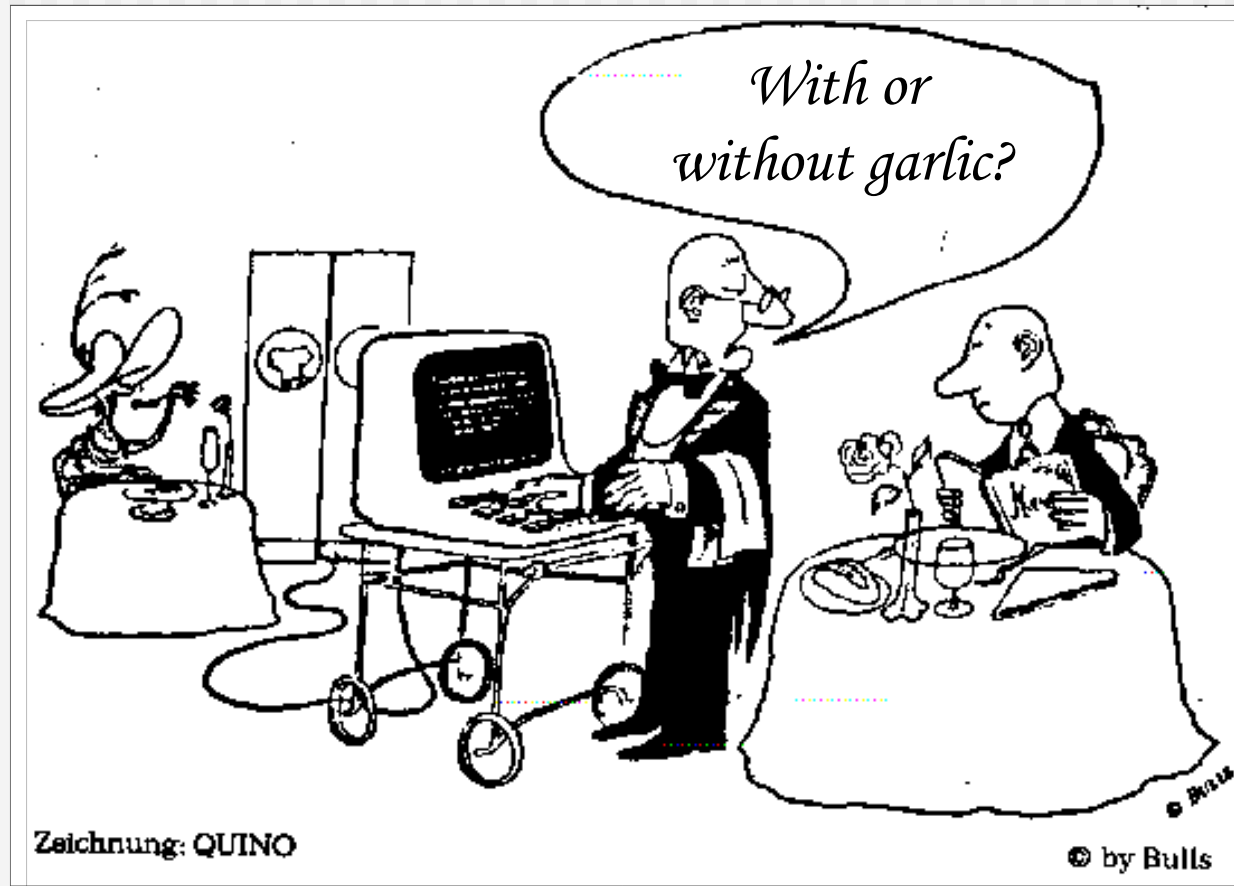
# Conclusion

---

- Good overview about the state of the art
- Interesting modifications to improve already existent solutions
  
- FNNQ seems to be a straight-forward solution
- Increase result quality by using
  - Service Areas
  - Global object density estimate
  - Remote NN Queries
  - Multiple threads



# Digitalized Restaurant



<http://b.kutzler.com/bk/m-h-restaurant.gif>

# Bibliography/Images

- Thomas Schwarz, Markus Iofcea, Matthias Grossmann, Nicola Hönle, Daniela Nicklas, Bernhard Mitschang (2004). *On Efficiently Processing Nearest Neighbor Queries in a Loosely Coupled Set of Data Sources*. In Proceedings of the 12th ACM International Symposium on Advances in Geographic Information Systems, pp. 184–193. acm press.
- Nick Roussopoulos, Stephen Kelley, Frédéric Vincent (1995). *Nearest neighbor queries*. In Proceedings of the 1995 ACM SIGMOD international conference on Management of data, pp. 71-79. acm press.
- Apostolos Papadopoulos, Yanis Manolopoulos (1996). *Parallel processing of nearest neighbor queries in declustered spatial data*. In Proceedings of the 4th ACM workshop of Advances in Geographic Information Systems, ACM-GIS 1996, pp. 35-43
  
- <http://www.snopes.com/common/lost/map.gif>
- <http://nvr9.com/private/postales/postales/Lost-Without-You.gif>
- [http://www.onno.groesz.com/gallery/skyscraper\\_B\\_02\\_winter.gif](http://www.onno.groesz.com/gallery/skyscraper_B_02_winter.gif)
- <http://www.onno.groesz.com/gallery/skyscraper.gif>
- [http://www.veiglwagenhuber.com/i\\_foto/mc.gif](http://www.veiglwagenhuber.com/i_foto/mc.gif)
- <http://www.pullover.de/groessen/meter.gif>
- <http://www.graefendorf.de/assets/images/fragezeichen.gif>
- <http://b.kutzler.com/bk/m-h-restaurant.gif>