Organic Computing
Peer-to-Peer-Netzwerke

Rolf Wanka

Sommersemester 2008

rwanka@cs.fau.de

Inhalte

- Kurze Geschichte der Peer-to-Peer-Netzwerke
- Das Internet: Unter dem Overlay
- Die ersten Peer-to-Peer-Netzwerke
  - Napster
  - Gnutella
- CAN (Content Addressable Network)
- Chord
- Pastry und Tapestry
- Gradoptimierte Netzwerke
  - Viceroy
  - Distance-Halving
  - Koorde
- Netzwerke mit Suchbäumen
  - Skipnet und Skip-Graphs
  - P-Grid
- Selbstorganisation
  - Pareto-Netzwerke
  - Zufallsnetzwerke
  - Selbstorganisation
  - Metrikbasierte Netzwerke Sicherheit in Peer-to-Peer-Netzwerken
- Anonymität
- Datenzugriff: Der schnellere Download
- Peer-to-Peer-Netzwerke in der Praxis
  - eDonkey
  - FastTrack
  - Bittorrent
- Peer-to-Peer-Verkehr
- Juristische Situation

Zwei Fragen zur Informationsfindung

- Wo ist es?
- Wie dorthin kommen?

- Napster:
  - Wo?
    - Auf dem Server ☺
  - Wie dorthin?
    - Zum Serverstau ☹
- Gnutella
  - Wo?
    - Weiss nicht ☹
  - Wie dorthin?
    - Alle fragen ☹

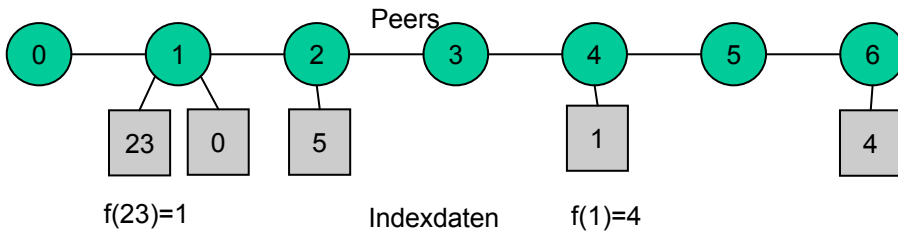
- Besser:

- Wo ist Datum x?
  - An der Stelle f(x)
  - Was ist f(x)?
    - Eine allen Teilnehmern bekannte Abbildung von x auf einem Raum

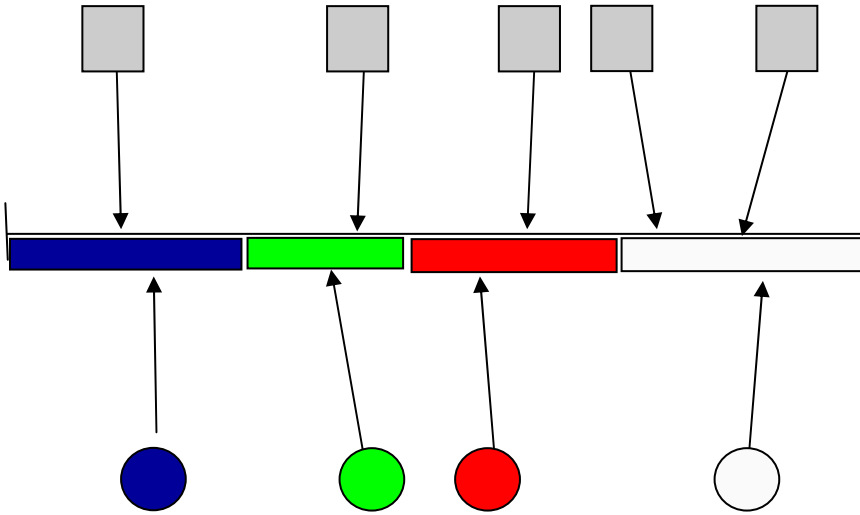
- Wie komme ich dorthin?
  - Durch eine Route die mir den Weg von meinen Standort zu f(x) aufzeigt.

Eigenschaften DHT

- Vorteile
  - Jedes Datum kann einem bestimmten Peer zugewiesen werden
  - Einfügen und Entfernen von Peers erzeugt nur Veränderungen in den benachbarten Peers



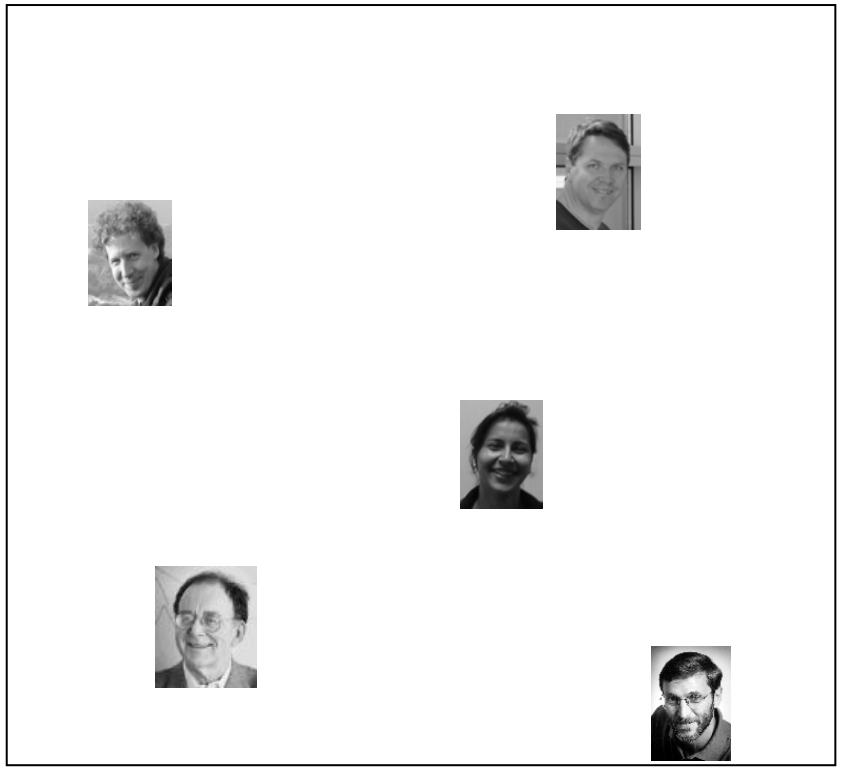
- DHTs werden von vielen P2P-Netzwerken benutzt



- Noch zu klären:
  - Die Verbindungsstruktur

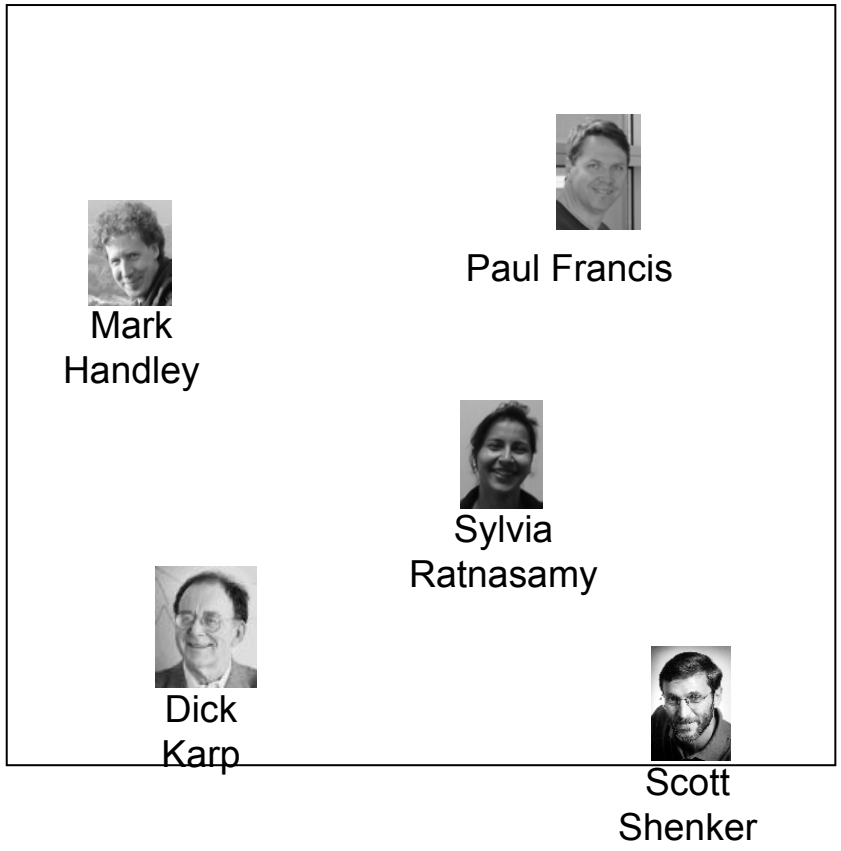
Content Addressable Network (CAN)

- Daten werden durch (zweiwertige, oder allgemein: *d*-wertige)-Hash-Funktion in das Quadrat abgebildet



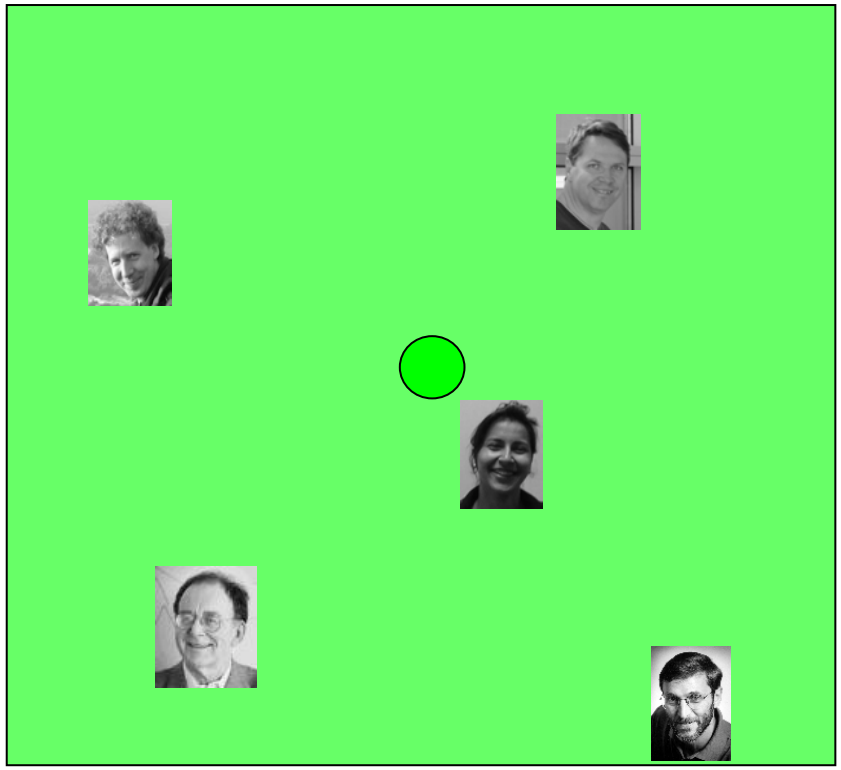
A Scalable Content Addressable Network (CAN)

- Daten werden durch (zweiwertige)-Hash-Funktion in das Quadrat abgebildet



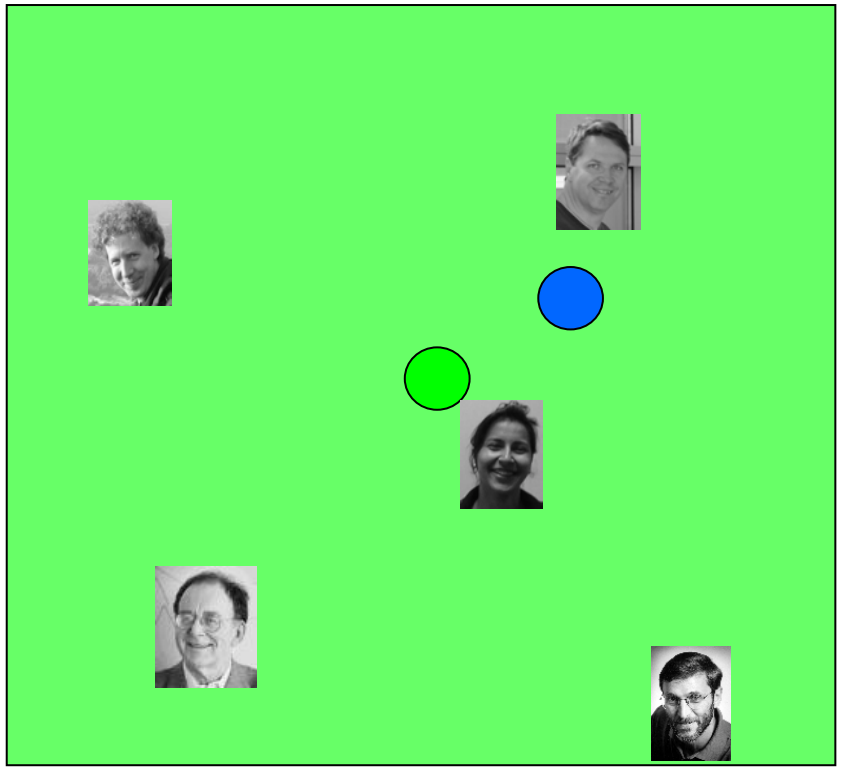
Content Addressable Network (CAN)

- Daten werden durch (zweiwertige)-Hash-Funktion in das Quadrat abgebildet.
- Am Anfang liegt ein leeres Quadrat mit nur einem Peer als Besitzer vor.



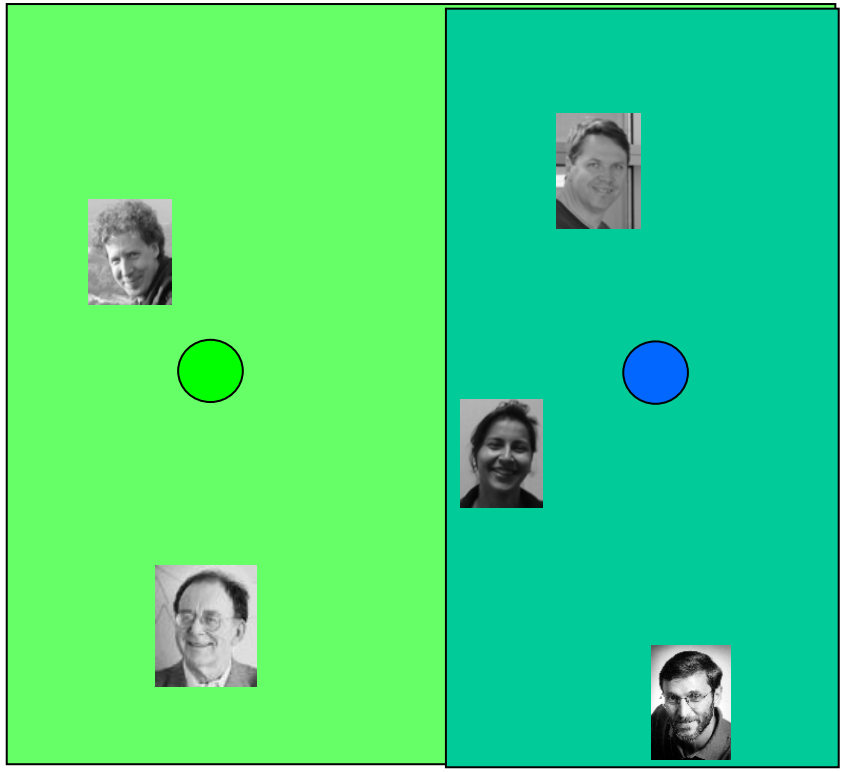
Content Addressable Network (CAN)

- Daten werden durch (zweiwertige)-Hash-Funktion in das Quadrat abgebildet
- Am Anfang liegt ein leeres Quadrat mit nur einem Peer als Besitzer vor.
- Der Besitzer einer Fläche speichert alle Einträge in der Fläche
- Ein Peer wählt einen zufälligen Punkt in der Ebene



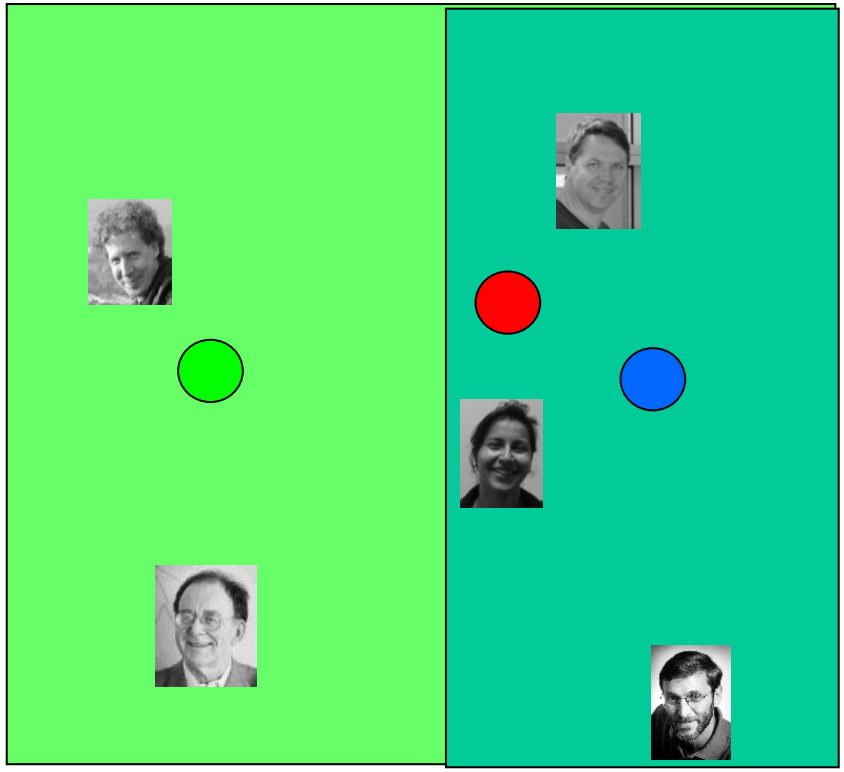
Content Addressable Network (CAN)

- Daten werden durch (zweiwertige)-Hash-Funktion in das Quadrat abgebildet
- Am Anfang liegt ein leeres Quadrat mit nur einem Peer als Besitzer vor.
- Der Besitzer einer Fläche speichert alle Einträge in der Fläche
- Ein Peer wählt einen zufälligen Punkt in der Ebene
  - Der Besitzer des entsprechenden Quadrats teilt seine Fläche und
  - übergibt die Hälfte dem neuen Peer



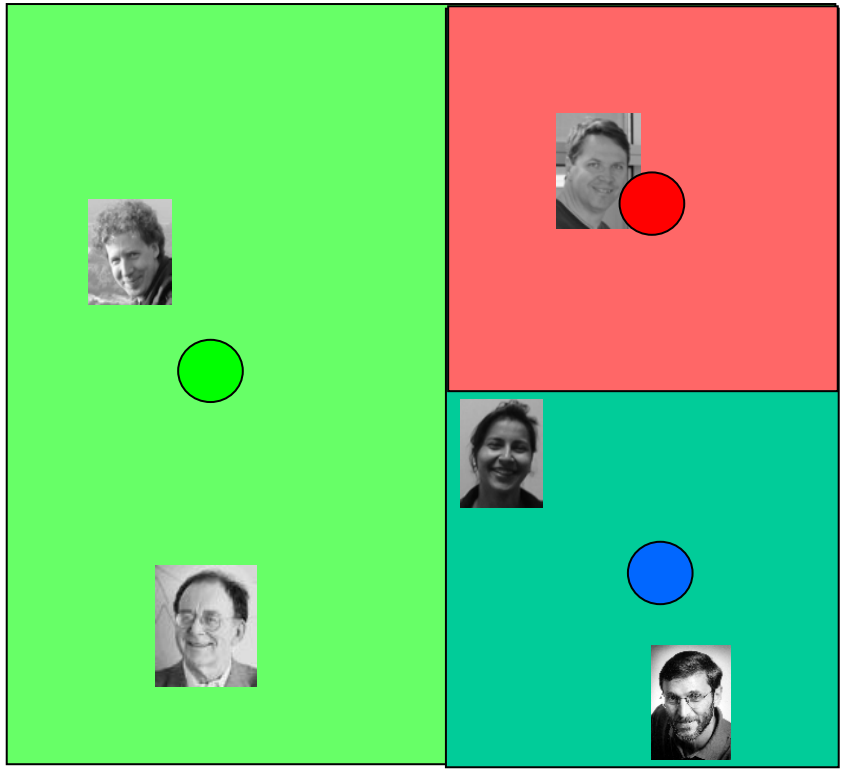
Content Addressable Network (CAN)

- Daten werden durch (zweiwertige)-Hash-Funktion in das Quadrat abgebildet
- Am Anfang ist ein leeres Quadrat mit nur einem Peer als Besitzer
- Der Besitzer einer Fläche speichert alle Einträge in der Fläche
- Ein Peer wählt einen zufälligen Punkt in der Ebene
  - Der Besitzer des entsprechenden Quadrats teilt seine Fläche und
  - übergibt die Hälfte dem neuen Peer



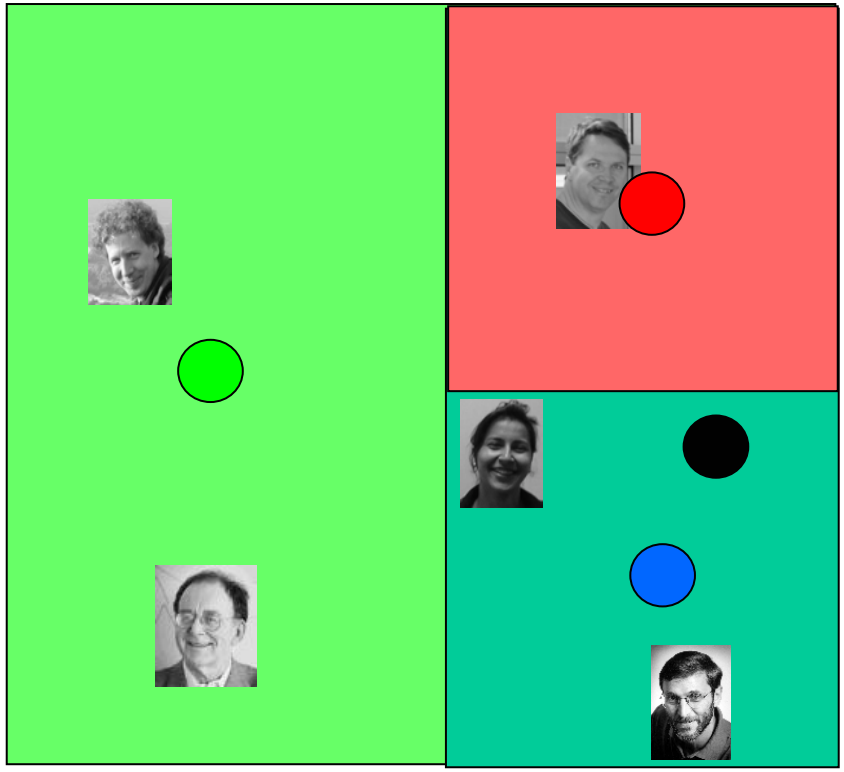
Content Addressable Network (CAN)

- Daten werden durch (zweiwertige)-Hash-Funktion in das Quadrat abgebildet
- Am Anfang liegt ein leeres Quadrat mit nur einem Peer als Besitzer vor.
- Der Besitzer einer Fläche speichert alle Einträge in der Fläche
- Ein Peer wählt einen zufälligen Punkt in der Ebene
  - Der Besitzer des entsprechenden Quadrats teilt seine Fläche und
  - übergibt die Hälfte dem neuen Peer



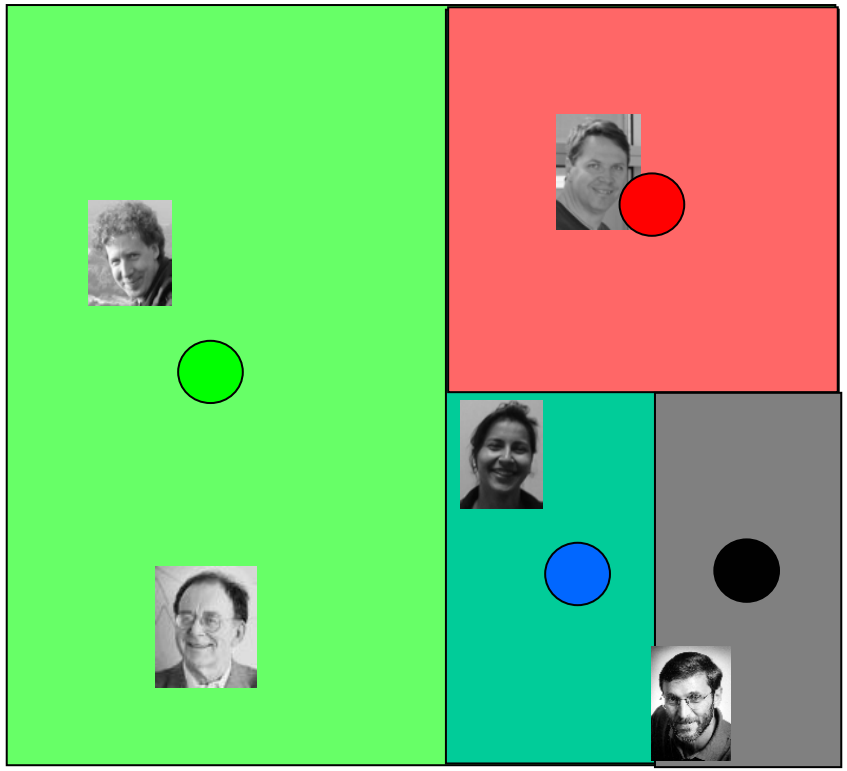
Content Addressable Network (CAN)

- Daten werden durch (zweiwertige)-Hash-Funktion in das Quadrat abgebildet
- Am Anfang liegt ein leeres Quadrat mit nur einem Peer als Besitzer vor.
- Der Besitzer einer Fläche speichert alle Einträge in der Fläche
- Ein Peer wählt einen zufälligen Punkt in der Ebene
  - Der Besitzer des entsprechenden Quadrats teilt seine Fläche und
  - übergibt die Hälfte dem neuen Peer



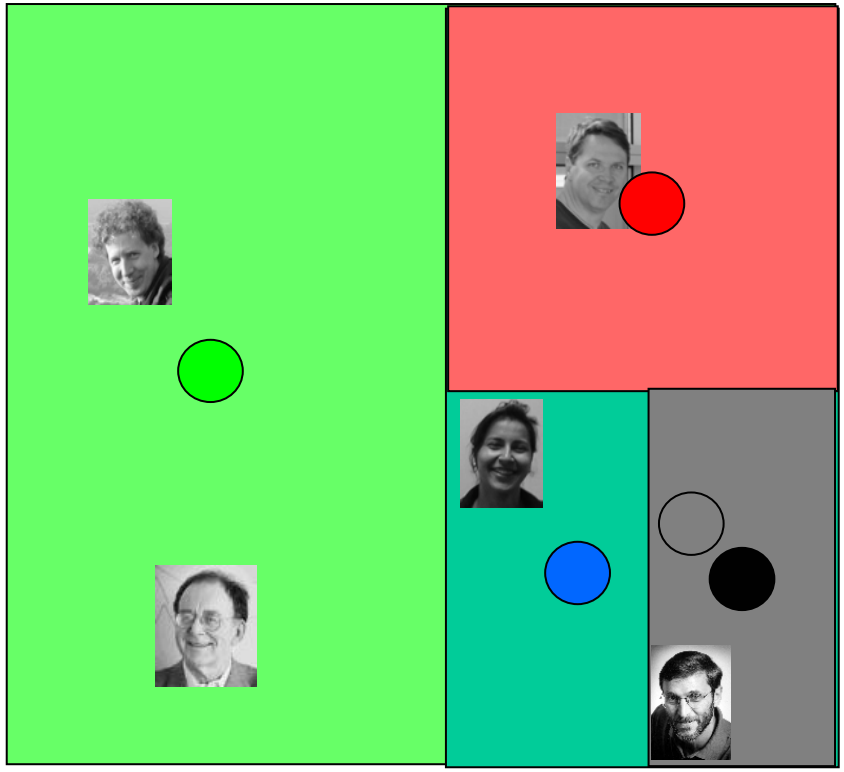
Content Addressable Network (CAN)

- Daten werden durch (zweiwertige)-Hash-Funktion in das Quadrat abgebildet
- Am Anfang liegt ein leeres Quadrat mit nur einem Peer als Besitzer vor.
- Der Besitzer einer Fläche speichert alle Einträge in der Fläche
- Ein Peer wählt einen zufälligen Punkt in der Ebene
  - Der Besitzer des entsprechenden Quadrats teilt seine Fläche und
  - übergibt die Hälfte dem neuen Peer



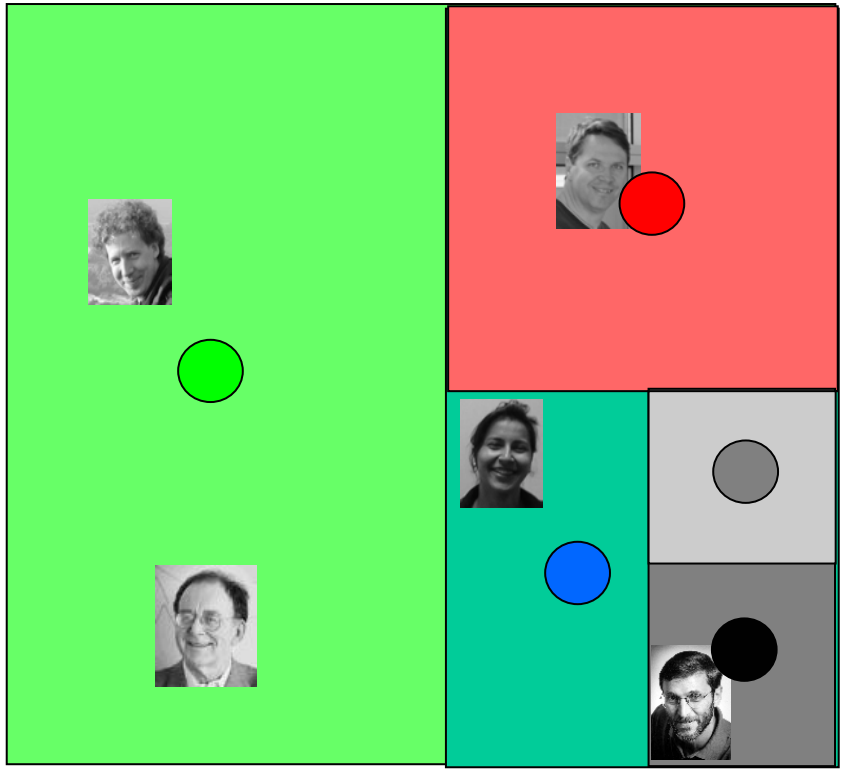
Content Addressable Network (CAN)

- Daten werden durch (zweiwertige)-Hash-Funktion in das Quadrat abgebildet
- Am Anfang liegt ein leeres Quadrat mit nur einem Peer als Besitzer vor.
- Der Besitzer einer Fläche speichert alle Einträge in der Fläche
- Ein Peer wählt einen zufälligen Punkt in der Ebene
  - Der Besitzer des entsprechenden Quadrats teilt seine Fläche und
  - übergibt die Hälfte dem neuen Peer



Content Addressable Network (CAN)

- Daten werden durch (zweiwertige)-Hash-Funktion in das Quadrat abgebildet
- Am Anfang liegt ein leeres Quadrat mit nur einem Peer als Besitzer vor.
- Der Besitzer einer Fläche speichert alle Einträge in der Fläche
- Ein Peer wählt einen zufälligen Punkt in der Ebene
  - Der Besitzer des entsprechenden Quadrats teilt seine Fläche und
  - übergibt die Hälfte dem neuen Peer



Wie groß/klein können solche Flächen werden?

R(p) : Rechteck eines Peers *p*

A(p) : Fläche von *R(p)*

n : Anzahl Peers

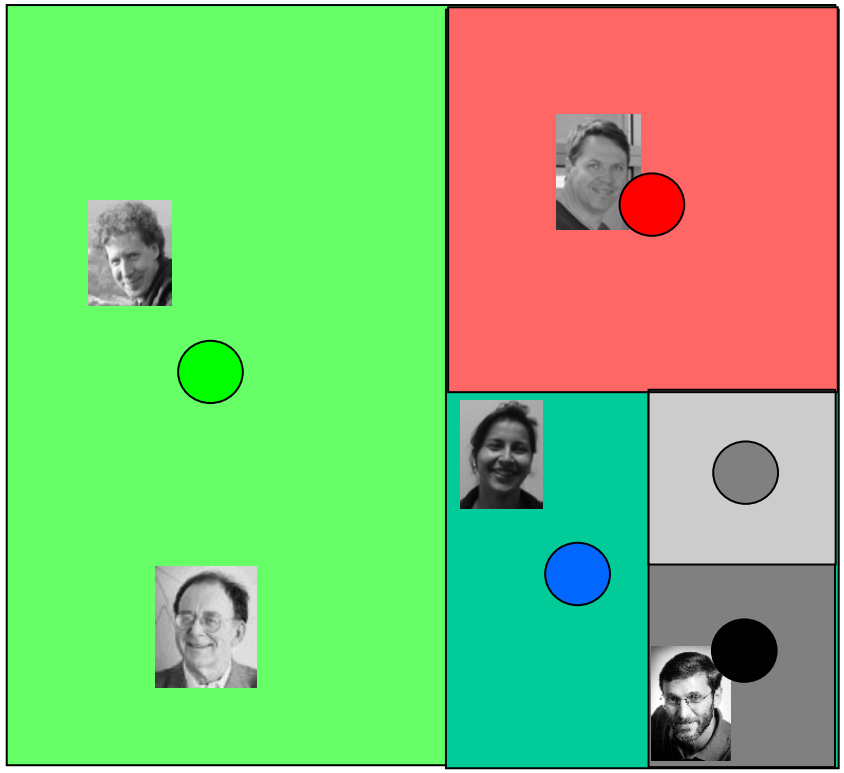
Anfangsquadrat: Fläche 1

Lemma:

Für alle Peers *p* gilt:

1. $E[A(p)] = \frac{1}{n}$
2. Sei $P_{R,n}$ die W'keit, daß keiner der *n* Peers in das (beliebige, aber feste) Rechteck *R* hineinfällt. Dann gilt:

$$P_{R,n} \leq e^{-n\mathrm{Vol}(R)}$$

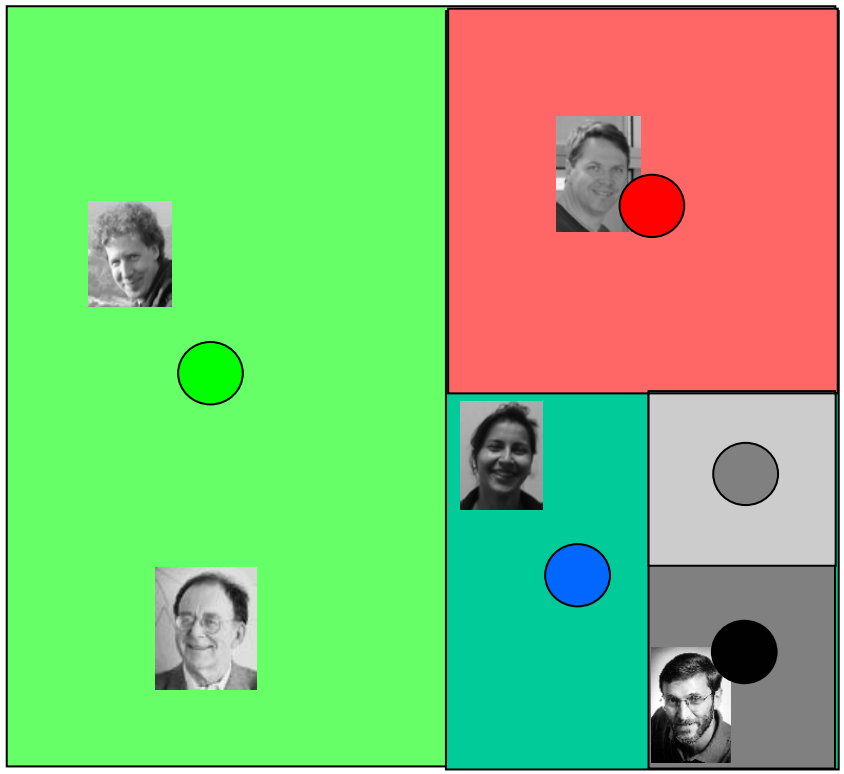


Die erwartete Fläche eines Peers in CAN

Beweis von 1: $E[A(p)] = \frac{1}{n}$

Seien {1,..,*n*} die Peers.

Dann gilt: $\sum_{i=1}^{n} A(i) = 1$



Ferner gilt wegen Symmetrie

$$\forall i \in \{1, \ldots, n\} \ : \ A(i) = A(1)$$

Damit gilt:

$$1 = \sum_{i=1}^{n} A(i) = E\left[\sum_{i=1}^{n} A(i)\right] = \sum_{i=1}^{n} E[A(i)] = nE[A(1)]$$

Ein nicht getroffenes Rechteck

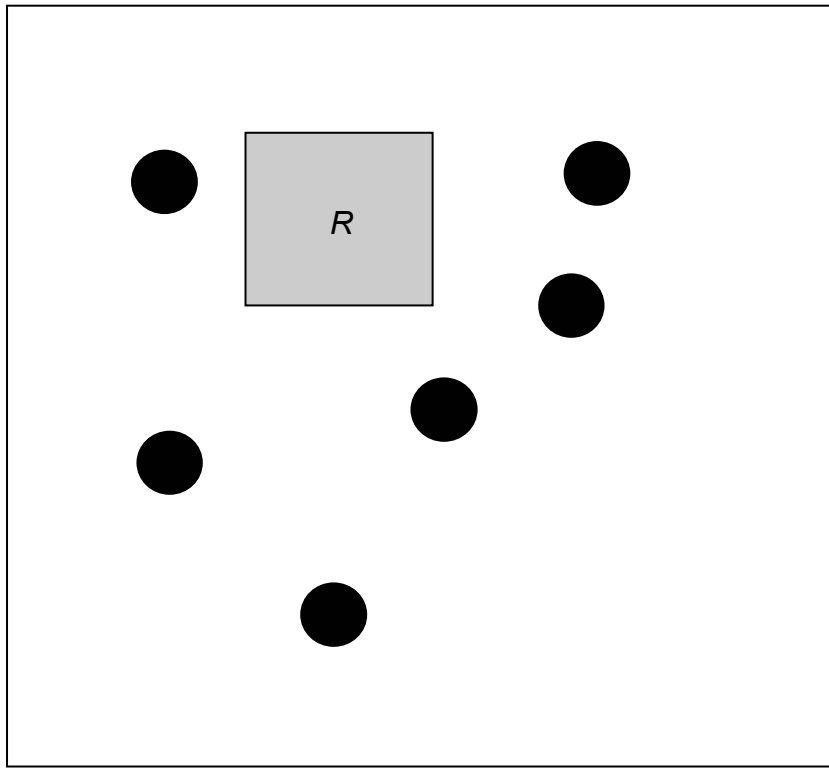
Beweis von 2: $P_{R,n} \leq e^{-n\mathrm{Vol}(R)}$

Betrachte ein Rechteck *R* der Fläche

x = Vol(*R*)

Die Wahrscheinlichkeit, daß ein Peer nicht in diese Fläche fällt, ist: $1 - x$

Die Wahrscheinlichkeit, daß *n* Peers nicht in *R* hineinfallen, ist: $(1-x)^n$



Also:

$$(1-x)^n = ((1-x)^{\frac{1}{x}})^{nx} \leq e^{-nx}$$

da für alle $m > 1 : \left(1 - \frac{1}{m}\right)^m \leq \frac{1}{e}$

Wie groß kann ein nicht getroffenes Rechteck sein?

Aus 2: $P_{R,n} \leq e^{-n\mathrm{Vol}(R)}$

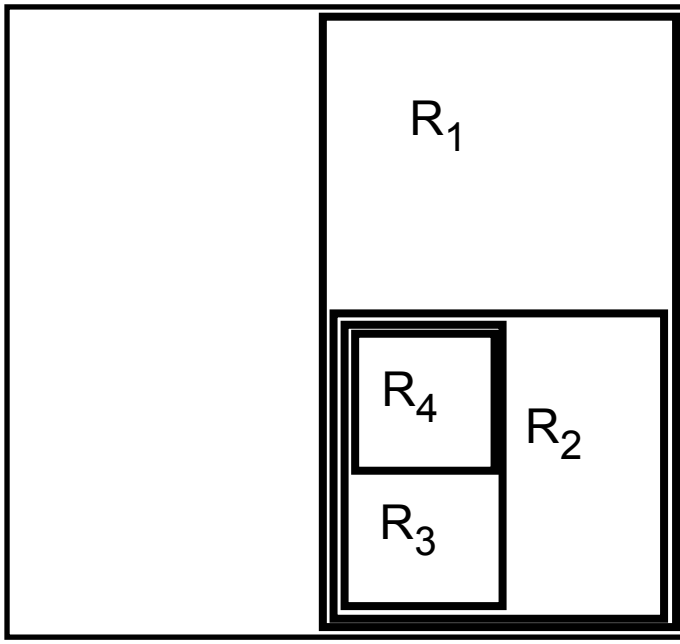
folgt für ein Rechteck R_i der Fläche $(1/2)^i$

$$P_{R_i, c2^i \ln n} \leq e^{-c2^i \ln n \mathrm{Vol}(R_i)} = n^{-c}$$

Es genügen also $c \cdot 2^i \cdot \ln n$ Peers, um R_i mit Wahrscheinlichkeit 1- n^{-c} zu teilen („w.h.p“).

Diese kommen nun in Schüben zu $c \cdot 2^i \cdot \ln n$

Sei nun $i \in \left\{1, \ldots, \log \frac{n}{2c \ln n}\right\}$



$$\sum_{i=1}^{\log \frac{n}{2c \ln n}} c \cdot 2^i \ln n = c \cdot (\ln n) \cdot \sum_{i=1}^{\log \frac{n}{2c \ln n}} 2^i \leq c \cdot (\ln n) 2^{\log \frac{n}{c \ln n}} = n$$

Damit wird ein Rechteck der Fläche $\frac{2c \ln n}{n}$ mit W‘keit $n^{-c} \log n$ nicht geteilt.

Wie gleichmäßig werden die Daten verteilt?

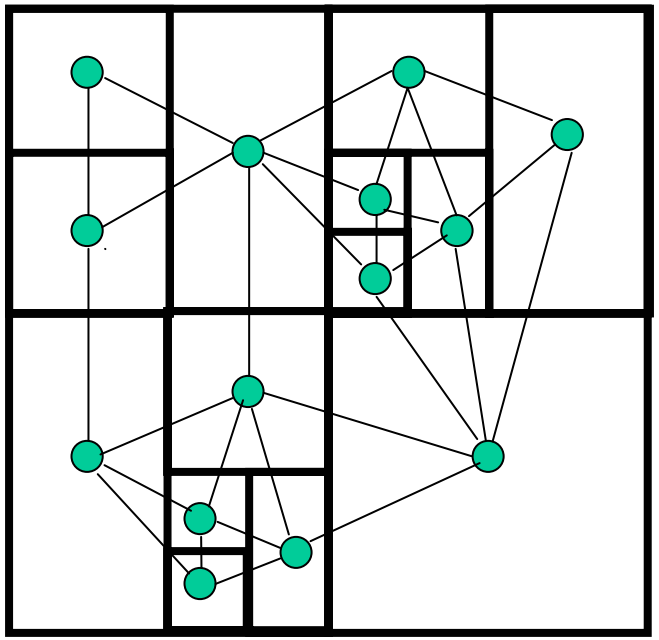
- Also:
  - Mit Wahrscheinlichkeit n (log n) $n^{-c} \leq n^{-c+2}$ wird ein Rechteck der Größe $2c(\ln n)/n$ bei n Peers nicht geteilt.

- Wenn m Daten insgesamt gespeichert werden,
  - so erhält jeder Peer also maximal $2\ c\ (\ln n)\ m/n$ Elemente,
  - während der Durchschnitt m/n Elemente speichert

- Also speichert jeder Peer höchstens $2c$ (ln n) Mal mehr als der Durchschnittspeer mit hoher Wahrscheinlichkeit („w.h.p.“).

Lookup in CAN

- Verbindungsstruktur: Zwischen den Besitzern horizontal und vertikal benachbarter Rechtecke bestehen Kanten.
- Zu Beginn der Suche: Der Ort des gesuchten Datums wird durch Berechnung der Hash-Wertes bestimmt
- Anfrage wird in Richtung des Ortes weitergeleitet
- *d* Dimension des „Quadrats“
  - 1: Linie
  - 2: Quadrat
  - 3: Würfel
  - 4: ...
- Erwartete Anzahl Hops in *d* Dimensionen: $n^{1/d}$
- Durchschnittlicher Grad eines Knotens: $O(d)$



Ende